UNIVERSIDAD DE SANTIAGO DE CHILE

FACULTAD DE CIENCIA

Departamento de Matemática y Ciencia de la Computación

Identificación óptica de compuestos orgánicos usando clasificación Random Forest

Felipe Osorio Urzúa

Profesor Guía: Felipe Herrera Urbina

Tesis para optar al título de Analista en Computación Científica

Santiago - Chile

2022

RESUMEN

En este trabajo se presenta un enfoque para la identificación basada en la respuesta óptica de un láser, utilizando técnicas de aprendizaje automático, construidas desde información de literatura científica. Se propone un modelo de aprendizaje automático basado en Random Forest para clasificar los patrones de refracción de láser en diferentes materiales, dentro del espectro visible. Además, se explora la extracción de información de la literatura científica relacionada con la óptica y la generación de datos sintéticos usando la ecuación de Sellmeier. La validación cruzada se utiliza para evaluar el rendimiento del modelo propuesto, obteniendo resultados de una precisión del 95 %, entrenado con el 20 % de lo datos. En el trabajo se propone una metodología que combina la tecnología de identificación por láser con la inteligencia artificial y la minería de datos, y puede ser aplicado en diversas áreas de la industria y la investigación.

Palabras clave: Aprendizaje automático; Detección láser; Generación sintética de datos

ABSTRACT

This work presents an approach for identification based on the optical response of a laser, using machine learning techniques built from scientific literature information. A machine learning model based on Random Forest is proposed to classify laser refraction patterns on different materials within the visible spectrum. In addition, we explore the extraction of information from scientific literature related to optics and the generation of synthetic data using the Sellmeier equation. Cross-validation is used to evaluate the performance of the proposed model, obtaining results with 95 % accuracy, trained with 20 % of the data. This work proposes a methodology that combines laser identification technology with artificial intelligence and data mining, and can be applied in various industries and research fields.
Keywords: Machine learning; Laser identification; Synthetic data generation

Tabla de contenidos

Índice de tablas

Índice de figuras

Introducción

La aplicación de técnicas de aprendizaje automático (*machine learning*) ha acelerado el análisis de datos científicos (Ratner *et al.*, 2019). Esto incluye la interpretación de datos, la identificación y el control de experimentos a través de instrumentos automatizados. Una aplicación posible de en esta área es la identificación autónoma de compuestos químicos, lo que podría impactar en el desarrollo de sensores compactos, portátiles y altamente precisos con costos de equipamiento más bajos: el aprendizaje automático es una herramienta habilitadora en el diseño de hardware para sistemas de sensores inteligentes. Los modelos de aprendizaje automático pueden ser entrenados con bases de datos experimentales para clasificar señales entrantes y predecir señales de salida que mejoren las capacidades de las plataformas de detección. A partir de esta información, se puede utilizar una estrategia de rediseño de sensores, impulsada por los datos, para reemplazar características irrelevantes de sistemas de identificación actuales y con posibilidad de mejorar iterativamente el rendimiento del sensor.

Actualmente, una forma de identificar compuestos químicos es lograda mediante caracterizaciones basadas en espectroscopia Raman, que usa la huella de absorción de la región infrarroja del espectro, involucrando peaks de absorción que son únicos para moléculas individuales, permitiendo así su identificación química. Al compararse con otros métodos de identificación como los métodos químicos, si bien son altamente precisos en su funcionamiento, en ciertas aplicaciones, las técnicas de detección óptica pueden ser más ventajosas debido a que la interacción entre la luz y la materia no causa daño alguno y puede ser procesada desde una distancia remota.

Las técnicas basadas en láser son prometedoras para la detección automatizada, debido a que la respuesta óptica de los materiales (índice de refracción) codifica suficiente información para la identificación de compuestos químicos (Bikku *et al.*, 2022), sin generar una interacción destructiva en ellos. La información del índice de refracción está fundamental-

mente relacionado con propiedades fisicoquímicas microscópicas como la polarizabilidad dinámica, así como con variables macroscópicas como la concentración, la temperatura y la presión (Mohan *et al.*, 2019).

Usando datos experimentales de índice refractivo de compuestos orgánicos puros y polímeros, extraídos desde repositorios y literatura científica sobre un amplio rango de frecuencias, desde el ultravioleta al infrarrojo, se desarrolló un clasificador usando modelos de machine-learning, para identificar especies orgánicas, basado en una medición de dispersión de la luz de onda única, en la región del espectro visible, lejos de resonancias de absorción.

La utilización de bases de datos espectroscópicas moleculares para entrenar algoritmos de aprendizaje automático en problemas de identificación química ha sido reportada, principalmente, enfocando los esfuerzos en entrenar clasificadores que utilizan los datos de absorción infrarroja y dispersión de la luz (Madden & Howley, 2009; Madden & Ryder, 2003; Park & Son, 2021). En general, este enfoque se debe a que la información que se obtiene de la región infrarroja corresponde casi a una “huella digital”, lo que permite discriminar entre compuestos.

El trabajo reciente de Bikku *et al.* (2022) muestra una prueba de concepto de un clasificador basado en el modelo Random Forest, inducido sobre datos de índice de refracción, en un amplio rango de longitudes de onda, el cual puede identificar compuestos químicos con un 98.1 % de precisión sobre la región visible del espectro.

Organización del documento

El escrito consta de cuatro capítulos que abordan diferentes aspectos del tema de investigación. En el Capítulo 1 se encuentra el Marco Teórico, el cual se inicia con una introducción al aprendizaje supervisado y la presentación de los antecedentes e historia de clasificación por árboles de decisión, seguido de una revisión de la literatura e implementaciones notables. Posteriormente, se abordan los conceptos fundamentales de métodos de aprendizaje por ensamble, para dar paso a la definición de modelo clasificador utilizado en el trabajo: “Random Forest”. Finalmente, se presentan definiciones clave relacionadas con el fenómeno óptico del cual se obtendrá la información para clasificar.

El Capítulo 2 es el marco metodológico, en el cual se describe el diseño experimental y se

justifica el enfoque metodológico utilizado. Se detallan las técnicas y materiales utilizados en el trabajo, incluyendo un breve resumen del orígen de los datos, descripción de técnicas OCR para extracción de datos, aumento de datos usando la ecuación de Sellmeier, diseño del conjunto de entrenamiento y muestreo para balance de este, para terminar con el ajuste de hiper-parámetros y una revisión de las métricas de clasificación relevantes para una clasificación.
En el Capítulo 3 se presentan los resultados obtenidos con la metodología propuesta y se realiza un análisis mediante una revisión del desempeño del modelo: la precisión general de la clasificación y los resultados obtenidos de la matriz de confusión.
El Capítulo 4 corresponde a las Conclusiones. Se abordan las implicaciones prácticas y teóricas de los resultados obtenidos, se discuten las limitaciones y se ofrecen recomendaciones para futuras investigaciones.

Objetivo general

Identificar entre compuestos químicos transparentes, usando como entrada datos de la respuesta óptica (índice de refracción), en la región del espectro visible.

Objetivos específicos

- Describir técnicas y modelos de Machine Learning utilizados.
- Describir tratamiento de datos para conjuntos de entrenamiento desbalanceados.
- Elaborar un conjunto de entrenamiento con datos relevantes al fenómeno óptico de estudio.
- Implementar un modelo clasificador con el conjunto de datos elaborado.
- Evaluar rendimiento de la clasificación.

Capítulo 1. Marco teórico

En este capítulo se presentará un resumen contextual relacionado con la teoría y definiciones relevantes al modelo de aprendizaje automático implementado, esto es, definir a grandes rasgos Aprendizaje Automático y Aprendizaje Supervisado (Sección 1.1), para posteriormente introducir el modelo de Árboles de Decisión (Sección 1.2). Más adelante, en la Sección 1.3, se expone una técnica para, desde un mismo conjunto de entrenamiento, inducir modelos de aprendizaje distintos entre sí, para posteriormente introducir el modelo de bosque aleatorio (Random Forest), usado en este trabajo. Al final del capítulo, en la Sección 1.4 se incluye una breve explicación del fenómeno óptico producido por la interacción de la luz con la materia.

1.1. Aprendizaje automático

El término *aprendizaje automático* o *machine learning* se refiere a una amplia gama de técnicas y algoritmos en los que una máquina puede extraer conocimiento a partir de datos observados (Hastie *et al.*, 2009; Kubat, 2017). Estas metodologías, desarrolladas hace casi 40 años, no deben considerarse obsoletas al comparar su desempeño con las técnicas más modernas de clasificación, como las redes neuronales en aprendizaje profundo. El análisis estadístico realizado sobre las implementaciones de estas técnicas "clásicas" ha optimizado y explorado los aspectos relevantes en su selección, habiendo sido evaluadas de manera exhaustiva a lo largo de los años.

La idea fundamental detrás de la tecnología del aprendizaje automático es emular el proceso natural de aprendizaje a partir de observaciones. Para lograr esto, se desarrollan diversas estrategias y modelos que permiten resolver problemas de clasificación, regresión, interpretación, recomendación y otros. Además de las heurísticas, también se emplean técnicas de codificación de datos para expresarlos en estructuras que simplifiquen su

procesamiento. Esta combinación permite a los modelos extraer información desde los datos y mejorar su rendimiento en tareas específicas a medida que se enfrentan a más datos.

Este trabajo, está enfocado principalmente en el modelo Random Forest, el cual será introducido más adelante, un modelo ampliamente utilizado para las tareas de clasficación y regresión que ha demostrado su eficacia y robustez en numerosas aplicaciones (Bikku *et al.*, 2022; Madden & Howley, 2009; Madden & Ryder, 2003; More & Rana, 2017; Park & Son, 2021; Swets, 1988). Exploraremos su aplicación en el contexto actual, destacando sus fortalezas y áreas de aplicación más adecuadas. A través de un análisis riguroso, examinaremos cómo estas técnicas han evolucionado y cómo siguen siendo relevantes en comparación con los enfoques más modernos.

1.1.1. Aprendizaje supervisado

El aprendizaje supervisado ocurre cuando a partir de los datos observados, una máquina "aprende" una función que mapea los valores de entrada en las salidas observadas (Kubat, 2017). En este tipo de aprendizaje, los datos utilizados para construir los modelos pertenecen a un conjunto denominado **conjunto de entrenamiento**, el que contiene las características de las observaciones y los valores de salida, representados usualmente por la letra y.

Se define un vector de características como la matriz columna X de la Ecuación (1.1).

$$X = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix}. \tag{1.1}$$

Cada componente del vector X representa a un valor de las características observadas. Las dimensiones $\{x_1, x_2, ..., x_n\}$ conforman el conjunto de características y usualmente representan aspectos relevantes del modelo de datos del problema en estudio. Son válidos valores categóricos o numéricos y en numerosas ocasiones puede hacerse uso de técnicas para definir nuevos elementos en el conjunto, creando características en función de otras ya existentes: por ejemplo, dados $x_1 : base$, $x_2 : altura$, se puede definir $x_3 : volumen = x_1 \cdot x_2$.

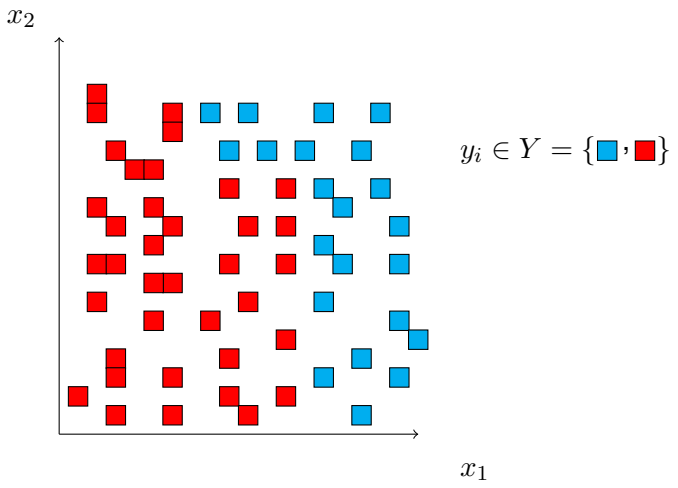


Figura 1.1: Conjunto de entrenamiento. Elaboración propia.

Formalmente, dado un conjunto T con N representaciones de observaciones en pares entradas-salidas de la forma de la Ecuación (1.2)

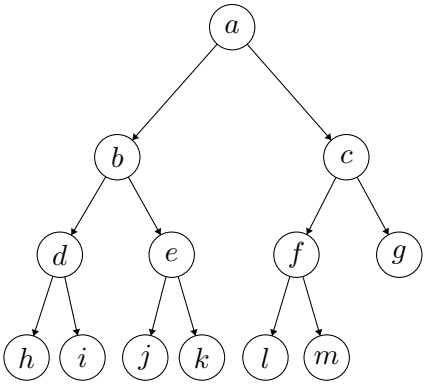
$$T = \{(X_1, y_1), (X_2, y_2), ..., (X_N, y_N)\}, \tag{1.2}$$

donde cada par es generado por una función existente no conocida $y = f(X)$, se busca una función h que aproxime a la función real f, de la forma de la Ecuación (1.3)

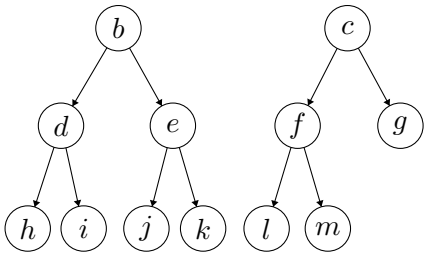
$$h(X_i) = y_i, \tag{1.3}$$

donde h corresponde al modelo inducido por los datos, X_i corresponde al vector de atributos o características e y_i el valor real, *target*, etiqueta verdadera o clase de esa observación. El conjunto Y corresponde al conjunto de posibles valores de y_i. En la Figura 1.1 se muestra un conjunto de entrenamiento, para las características x_1 y x_2 junto con su clase $y_i \in Y = \{\text{cían},\ \text{rojo}\}$.

Una de las razones por las que se prefiere diseñar sistemas que aprendan de los datos y no diseñar sistemas inteligentes que realicen la tarea en primer lugar, es que hay situaciones en las cuales los diseñadores de algoritmos no pueden anticipar todos los posibles escenarios a los que su sistema se enfrentará: un algoritmo de resolución de laberintos debe aprender antes cómo representar el laberinto que resolverá o un programa que predice valores en la bolsa, debe aprender día a día a interpretar los cambios en los datos. También hay situaciones donde ciertas tareas resultan mucho más sencillas usando las técnicas de aprendizaje automático: identificar un rostro es un problema relativamente simple si se resuelve utilizando algoritmos de aprendizaje automatizado.



(a) Árbol.



(b) Bosque.

Figura 1.2: Al remover el nodo raíz de un *árbol*, se obtiene un conjunto de árboles desconectados. Un conjunto con dos o más árboles separados se denomina *bosque*. Elaboración propia.

Existen distintas técnicas de aprendizaje supervisado para inducir la función h de la Ecuación (1.3), a continuación se ahondará en los modelos de interés para este trabajo.

1.2. Árboles de decisión

En esta sección se caracteriza el árbol como grafo, para posteriormente presentar los árboles de decisión y la formalización de un modelo de aprendizaje automático.

1.2.1. Árboles como estructura de datos

En la ciencia de la computación, una estructura de datos no lineal cuyo estudio es fundamental es el árbol enraizado (Levitin, 2012) (en adelante, simplemente árbol), el cual se caracteriza por ser un grafo dirigido, donde cada par de vértices del árbol se conecta exactamente una vez.

Es debido a esta única conexión que, si se selecciona arbitrariamente cualquier vértice del árbol, este vértice seleccionado puede ser considerado raíz, lo que resulta de conveniencia al diseñar y analizar algoritmos recursivos. Usualmente la raíz de un árbol se ubica arriba, considerando este vértice como *nivel 0 del árbol*[1], los vértices ubicados abajo de él se consideran de nivel 1 y subsecuentes niveles para los vértices ubicados abajo de ellos. Un vértice sin descendientes se denomina como hoja. Entre las numerosas aplicaciones de

[1] Algunos autores lo designan como nivel 1.

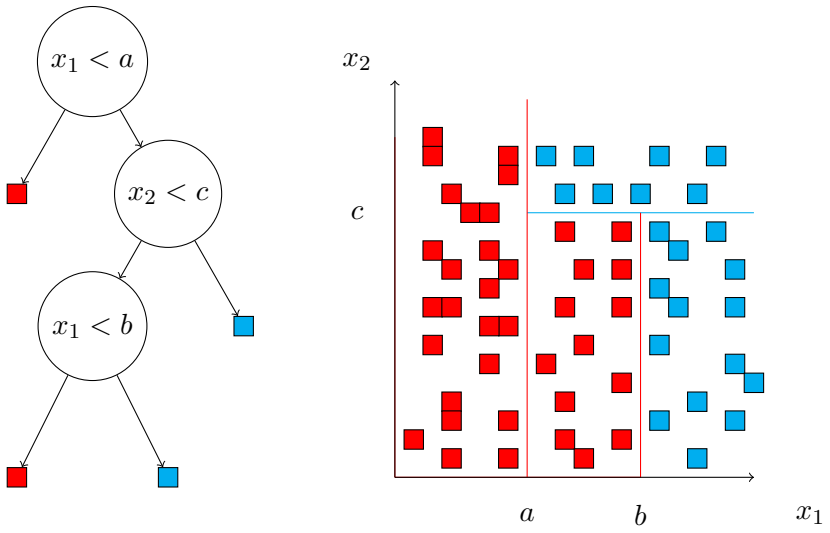
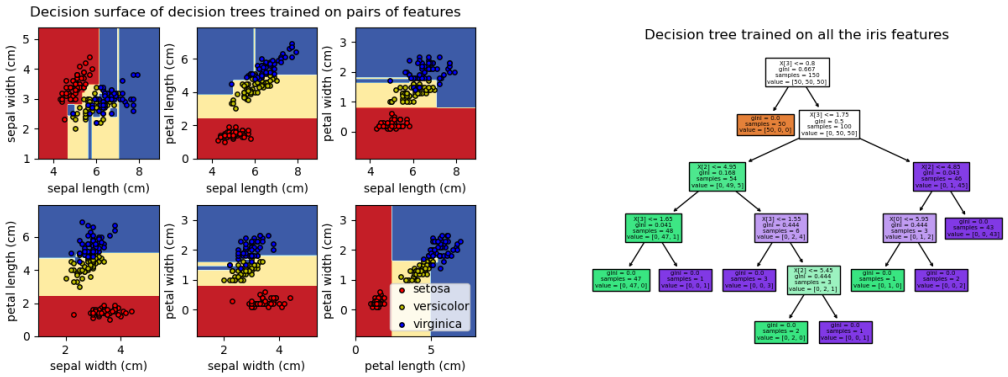


Figura 1.3: División del espacio característica al separar según valores para cada atributo. Cada vértice interno representa una condición que será evaluada por el árbol. Cada evaluación representa una línea que divide en la dimensión x_i en dos regiones. Las hojas son las clases objetivo. Elaboración propia.

los árboles, se incluyen representaciones jerárquicas, como árboles genealógicos, árboles de sistemas de archivos o árboles binarios de búsqueda. Usualmente al referirse a un vértice se usa indistintamente la palabra nodo. En este trabajo se estudiará la aplicación de árboles de decisión en machine learning, usando el modelo conocido como *C4.5* (Quinlan, 1986).

1.2.2. Árboles de Decisión como clasificadores

En el contexto de aprendizaje automático, se definen los árboles de decisiones como modelos que realizan predicciones evaluando una serie de pruebas (por simplicidad, binarias), en un punto de datos del conjunto de entrenamiento T (ver Ecuación (1.2)). Dichas pruebas están convenientemente representadas dentro de la estructura de cada vértice o nodo del árbol, realizando una comparación de un atributo $x_i \in X$ para un valor v en cada uno de ellos, descendiendo hasta las hojas donde se realiza la clasificación.
El espacio característica es recursivamente dividido por estas pruebas en una forma rectangular (ver Figura 1.3). Se puede observar que, dada la suficiente cantidad de divisiones y siempre que dos puntos que coincidan no tengan etiquetas diferentes, este tipo de clasificador puede correctamente etiquetar cualquier conjunto de entrenamiento. Otro ejemplo de división del espacio característica puede ser visto en la Figura 1.4.



(a) Iris Dataset.

(b) Árbol decisión resultado.

Figura 1.4: (a) División del espacio característica bidimensional, para pares de características del conjunto de entrenamiento Iris. En (b) un árbol de decisión ”entrenando“ sobre todas las características del conjunto: los nodos internos evalúan diferentes valores de x_1, x_2 y x_3 para realizar una clasificación. Tomada de scikit-learn 1.1.2 documentation. (s.f.). Plot the decision surface of decision trees trained on the iris dataset. https://scikit-learn.org/stable/auto_examples/tree/plot_iris_dtc.html

Construir un árbol binario de decisión optimal es un problema NP-completo (Hyafil & Rivest, 1976): diversas heurísticas se han propuestos en la construcción de árboles binarios de decisión cercanos al más óptimo. El algoritmo 1, adaptado de (Kubat, 2017), toma la idea del artículo de Quinlan (1986) y proporciona un pseudocódigo para inducir el crecimiento de árboles de decisión, asumiendo que existe un método de cuantificar cuanta información provee cada atributo. Buscar la característica x_i de X que aporte la mayor cantidad de información para luego considerar solo los x_i valores como candidatos para realizar las particiones. Se busca que el clasificador realice divisiones que aumenten la certeza de a qué clase pertenece cada punto. El caso ideal es cuando el clasificador encuentra un plano (o hiper-plano, para más dimensiones) $x_i = v$ que separe todos los elementos del conjunto de entrenamiento tal que todas las observaciones que pertenezcan a una clase y_i estén en un lado y los que no, estén en el otro lado de la división. Cuando todas las clases de un nodo coinciden (o están dentro de un margen de error) se dice que es un nodo puro y se crea una hoja, con la clase correspondiente.

Algoritmo 1: Inducción de árbol de decisión desde T. Adaptado de (Kubat, 2017)

Datos: Sea T el conjunto de entrenamiento

grow(T):

(1) Encontrar el atributo x_i y valor v (ver 1.1)que contribuye la **máxima información** acerca de las etiquetas de clase en T.

(2) Dividir T en T_j subconjuntos, caracterizados por diferentes valores de x_i (para un caso binario, separar entre datos menores al valor v o no).

(3) **para cada** T_j **hacer**

> **si** *Todos los elementos en* T_j *pertenecen a la misma clase* **entonces**
> Crear una hoja etiquetada con esta clase.
>
> **en otro caso**
> Aplicar recursivamente el procedimiento para cada subconjunto T_j (grow(T_j)).

1.2.3. Entropía e información

Para medir cuanta información entrega un atributo, se introduce el concepto de entropía de la información, como medida cuantitativa de incertidumbre para una variable aleatoria (Cover & Thomas, 2005). Sea Y una variable aleatoria discreta, la entropía de Y denotada por $H(Y)$ se define como la Ecuación (1.4):

$$H(Y) = -\sum_{k \in Y} P(Y = k) \cdot log_2 P(Y = k) \tag{1.4}$$

La suma de la Ecuación (1.4) corresponde a la cantidad de información y es funcional a la distribución de probabilidad $P(Y = k) \forall k \in Y$.

Por ejemplo, sea $Y = \{a, b, c, d\}$ el conjunto de posibles valores, con probabilidades de la Ecuación (1.5).

$$Y = \begin{cases} a, \text{ con probabilidad } \frac{1}{2}, \\ b, \text{ con probabilidad } \frac{1}{4}, \\ c, \text{ con probabilidad } \frac{1}{8}, \\ d, \text{ con probabilidad } \frac{1}{8}. \end{cases} \tag{1.5}$$

Utilizando la Ecuación (1.5) en la Ecuación (1.4), se calcula la entropía de Y en la

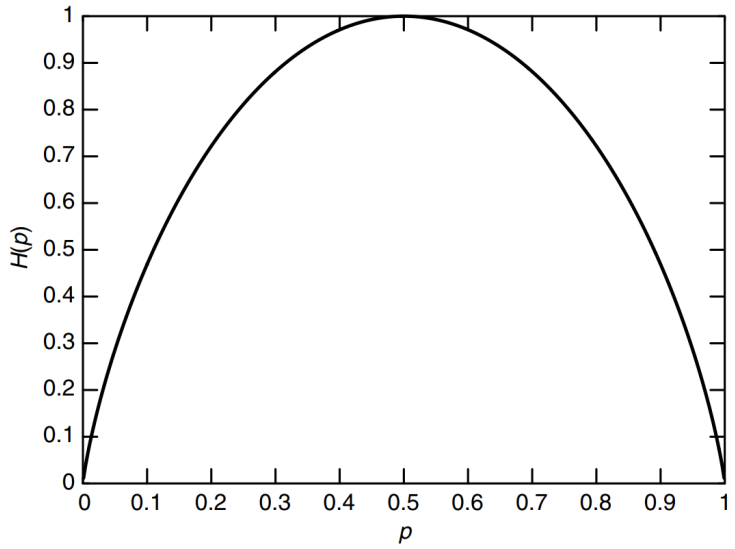


Figura 1.5: Entropía en función de la probabilidad p de una variable aleatoria con distribución de Bernoulli. Tomada de (Cover & Thomas, 2005).

Ecuación (1.6):

$$H(Y) = -\left(\underbrace{\frac{1}{2}\cdot log_2\frac{1}{2}}_{a} + \underbrace{\frac{1}{4}\cdot log_2\frac{1}{4}}_{b} + \underbrace{\frac{1}{8}\cdot log_2\frac{1}{8}}_{c} + \underbrace{\frac{1}{8}\cdot log_2\frac{1}{8}}_{d}\right) = \frac{7}{4} \text{ bit.} \tag{1.6}$$

Para el caso de una variable aleatoria con distribución de Bernoulli, donde $P(Y = p) = p$ y $P(Y = n) = 1 - p$. Un caso de interés es $p = \frac{1}{2}$, donde la distribución de probabilidad asociada a Y es la Ecuación (1.7)

$$Y = \begin{cases} p, \text{ con probabilidad } \frac{1}{2}, \\ n, \text{ con probabilidad } \frac{1}{2}. \end{cases} \tag{1.7}$$

Luego, la entropía de Y en la Ecuación (1.7) está dada por la Ecuación (1.8).

$$H(Y) = -\left(\underbrace{\frac{1}{2}\cdot log_2\frac{1}{2}}_{p} + \underbrace{\frac{1}{2}\cdot log_2\frac{1}{2}}_{n}\right) = 1 \text{ bit.} \tag{1.8}$$

Se observa en la Figura 1.5 que la entropía de la Ecuación (1.8) es máxima exactamente cuando $p = \frac{1}{2}$, que es el máximo valor posible para cada probabilidad.

La definición de la Ecuación (1.4) es válida para variables aleatorias discretas y esa la principal idea que se enlaza con la salida de un clasificador, ya que solo puede tomar

valores discretos dentro del conjunto de clases Y. Así, la probabilidad de que Y tome el valor k es la proporción de los datos que tienen la clase k, definido en la Ecuación (1.9).

$$P(Y = k) = \frac{|\{i|y_i = k\}|}{n} \tag{1.9}$$

Al inducir el árbol sobre el conjunto de entrenamiento, se busca minimizar la entropía en cada partición recursiva del conjunto y aumentar la pureza de cada nodo hasta realizar una clasificación.

A continuación, se define la variable indicador $X_{i,v}$ con valor 1 cuando $x_i < v$ y 0 en cualquier otro caso. Se introduce el concepto de entropía condicional, de Y dado un $X_{i,v}$ como la media ponderada por la entropía de ambos lados de la partición generada por $x_i < v$ (Figura 1.6), definido formalmente en la Ecuación (1.10).

$$H(Y|X_{i,v}) := P(X_{i,v} = 1)H(Y|X_{i,v} = 1) + P(X_{i,v} = 0)H(Y|X_{i,v} = 0). \tag{1.10}$$

De la misma manera, resulta conveniente definir la información mutua $I(X_{i,v}; Y)$ como la reducción en incertidumbre de $X_{i,v}$, debido al conocimiento de Y, en la Ecuación (1.11).

$$I(X_{i,v}; Y) := H(Y) - H(Y|X_{i,v}). \tag{1.11}$$

Esta cantidad es utilizada como medida de certeza de la partición, es siempre no-negativa y es cero solo cuando las proporciones en la distribución de etiquetas son iguales antes y después de la partición.

Considerando las expresiones de la Ecuación (1.4), Ecuación (1.10) y la Ecuación (1.11), se puede profundizar la idea inicial del Algoritmo 1, sintetizando un algoritmo que busque el atributo más informativo, descrito en el Algoritmo 2.

Como heurística, el algoritmo itera “vorazmente” sobre los puntos del conjunto de entrenamiento para buscar el mejor valor v que divida el espacio característica.

1.2.4. Impureza de Gini

El algoritmo CART para inducir árboles de decisiones de Breiman *et al.* (1984) también usa la idea de buscar particiones que minimicen la impureza de los nodos, con la diferencia que encuentra el atributo más informativo mediante el cálculo del índice de diversidad de Gini-Simpson (comúnmente llamado impureza de Gini), el cual es calculado de la forma

Algoritmo 2: Encontrar el Atributo Más Informativo (AMI). Adaptado de Kubat (2017)

Datos: Sea T el conjunto de entrenamiento

AMI(T):

(1) Calcular la entropía de T usando los porcentajes de las etiquetas como valores de probabilidad.

$$H(Y) = -\sum_{k \in Y} P(Y = k) \cdot log_2 P(Y = k)$$

(2) **para cada** $x_j \in X$ *que particiona* T *binariamente con el valor* v **hacer**

- (i) Calcular la entropía de cada partición T_i.
- (ii) Calcular la entropía condicional $H(Y|X_{i,v})$.
- (iii) Calcular la ganancia de información $I(X_{i,v}; Y) = H(Y) - H(Y|X_{i,v})$.

(3) Elegir el atributo con mayor valor de ganancia de información.

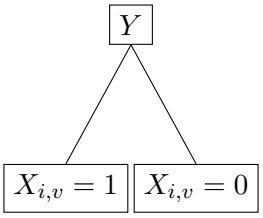


Figura 1.6: Partición generada en el conjunto de entrenamiento T por la evaluación de $x_j < v$. Cada subconjunto posee una entropía, dada por la distribución de sus etiquetas. Elaboración propia.

de la Ecuación (1.12).

$$G(Y) = \sum_{k} P(Y = k) \sum_{j \neq k} P(Y = j) \tag{1.12}$$

De manera análoga a la mostrada en la definición de la Ecuación (1.10), se puede definir una cantidad dependiente de la partición que genera $x_j < v$, como en la Ecuación (1.13).

$$G(Y|X_{j,v}) := P(X_{j,v} = 1)G(Y|X_{j,v} = 1) + P(X_{j,v} = 0)G(Y|X_{j,v} = 0) \tag{1.13}$$

La expresion de la Ecuación (1.13) puede ser utilizada para reducir la "impureza" de las particiones definidas recursivamente.

1.2.5. Criterios de detención

Para controlar la complejidad del árbol de decisión es necesario establecer criterios de detención y evitar que el tamaño del árbol sea tan grande que no quepa en la memoria. Usualmente la complejidad en espacio es controlado por alguna de las siguientes métricas:

número total de nodos, número total de hojas, altura del árbol y la cantidad de atributos usados. La fase de crecimiento continua hasta que se rompa alguna regla de detención como las que siguen:

1. Todas las instancias en en el nodo actual pertenecen a un mismo valor de y_i (i.e. un nodo puro).
2. La altura máxima del árbol ha sido alcanzada.
3. El número de casos en el nodo no cumple con los mínimos de casos que se necesitan para realizar una nueva división.
4. El criterio de partición no entrega ganancias superiores a un valor umbral.

Sin un criterio de detención apropiado, el proceso de selección de característica podría ejecutarse sin detención, ajustándose demasiado a los datos (overfitting) y perdiendo capacidad de generalización.

1.2.6. Implementaciones actuales

La idea de árboles de decisión para clasificación, según menciona (Quinlan, 1993) provienen desde fines de la década de 1950, su uso aparece en la literatura desde (Morgan & Sonquist, 1963). Las implementaciones de código abierto más actuales, que son ampliamente utilizadas, entre las que se incluye a scikit-learn en *Python* (Pedregosa *et al.*, 2011), *rpart* en R (Therneau *et al.*, 2015) o Weka para *Java* (Witten *et al.*, 2011), disponen de ambos criterios de partición presentados por (Breiman *et al.*, 1984) y (Quinlan, 1986) para la construcción de árboles de decisión.
Los árboles de clasificación examinan en profundidad los datos del conjunto de entrenamiento, esto entrega una mayor perspectiva de qué variables tienen mayor relevancia en la decisión final, más de lo que puede ser examinado por modelos más simples, como una regresión logística (Buchner *et al.*, 2017).

1.3. Métodos de aprendizaje ensamble

El aprendizaje por ensamble es una técnica de aprendizaje automático que ha ganado una gran popularidad en los últimos años debido a su capacidad para mejorar la precisión y la

estabilidad de las predicciones de los modelos individuales que componen el ensamble (Hakim *et al.*, 2021; Pes, 2020; Way *et al.*, 2012). Esta técnica se basa en la combinación de múltiples modelos más simples (modelos base) para producir una predicción más precisa y robusta. La combinación de múltiples modelos puede ayudar a reducir el riesgo de sobreajuste y mejorar la generalización de las predicciones a nuevos datos.
En esta sección, se revisará en detalle el método de aprendizaje por ensamble llamado 'Bagging' y cómo es utilizado en el modelo 'Random Forest', usando como modelo base árboles de decisiones, revisando los conceptos básicos detrás de esta técnica.

1.3.1. Bagging

La construcción de árboles de decisión es inestable, en particular para conjuntos de entrenamiento pequeños, donde considerar (o no) un punto de datos altera las regiones de partición. Bagging (Breiman, 1996) es un método de ensamble efectivo para contrarrestar esta desventaja. Es una técnica de muestreo del conjunto de entrenamiento, donde se selecciona aleatoriamente y con repetición, una muestra del conjunto original para construir un subconjunto de entrenamiento, el cuál será usado para construir uno de los modelos. Este proceso de muestreo es repetido para formar un ensamble de modelos que permitan predecir, mediante un mecanismo de agregación, una salida final.

1.3.2. Clasificador Random Forest

El modelo Random Forest (Breiman, 2001) lleva la idea de "Bagging" a un modelo de clasificación basado en un ensamble de árboles de decisión. Típicamente para clasificación, el método de agregación de Random Forest es por votación mayoritaria, pero otros mecanismos pueden ser utilizados. Su principal ventaja con respecto a un único árbol de decisión es la reducción de varianza promedio en la salida (ver Ecuación (1.14))

$$Var(\frac{1}{n}\sum_{i=1}^{n} Y_i) = (\frac{1}{n})^2 \sum_{i=1}^{n} Var(Y_i) = \frac{1}{n^2} \cdot n\sigma^2 = \frac{\sigma^2}{n} \tag{1.14}$$

sin afectar la salida esperada (ver Ecuación (1.15))

$$\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n} Y_i] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[Y_i] = \frac{1}{n} \cdot n\mu = \mu. \tag{1.15}$$

En la Figura 1.8 se ilustra un mecanismo de votación para un ensamble de árboles que decide sobre una observación de entrada.

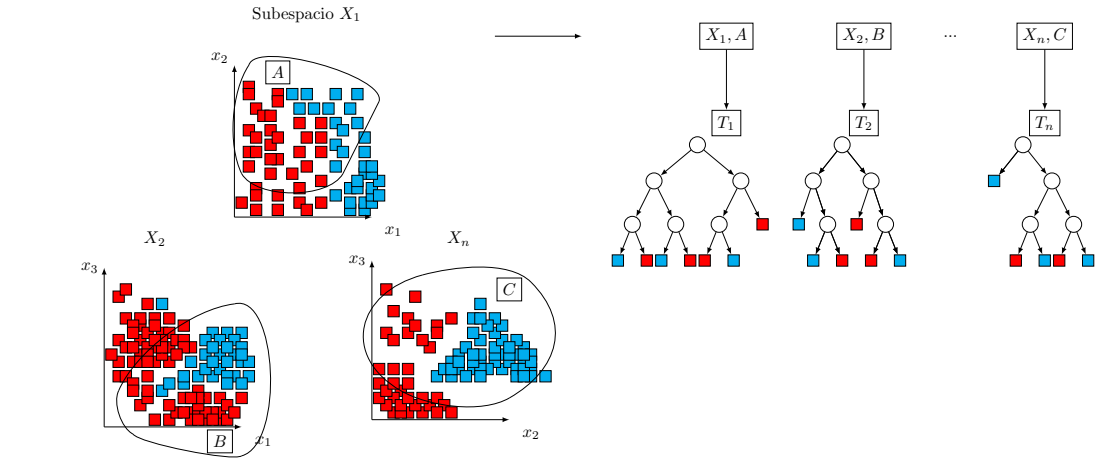


Figura 1.7: Bagging Trees. Las características $x_{\{1,...j\}}$ se seleccionan en subespacios $X_{\{1,...,n\}}$ de los cuales se obtienen las muestras (en la Figura A, B y C) que conforman el conjunto de entrenamiento para cada árbol $T_{\{1,...,n\}}$. Elaboración propia.

1.4. Detección óptica

Para lograr obtener un modelo que represente los rasgos más importantes de un fenómeno real, es necesario tener conocimiento del dominio de los datos que son observados de este fenómeno. A continuación se introduce el fenómeno óptico desde donde se obtienen datos utilizados para este trabajo.

1.4.1. Interacción entre radiación y materia

La naturaleza de las interacciones entre la radiación y la materia es ampliamente descrito en la literatura (Gründler, 2007; Steen & Mazumder, 2010) y se puede clasificar en dos tipos de interacciones: una interacción inelástica con pérdida de energía, y sin pérdida de energía por una interacción elástica. Las interacciones elásticas entre radiación y materia, como la reflexión o la refracción revelan información de las composición molecular de los medios por los cuales viaja.

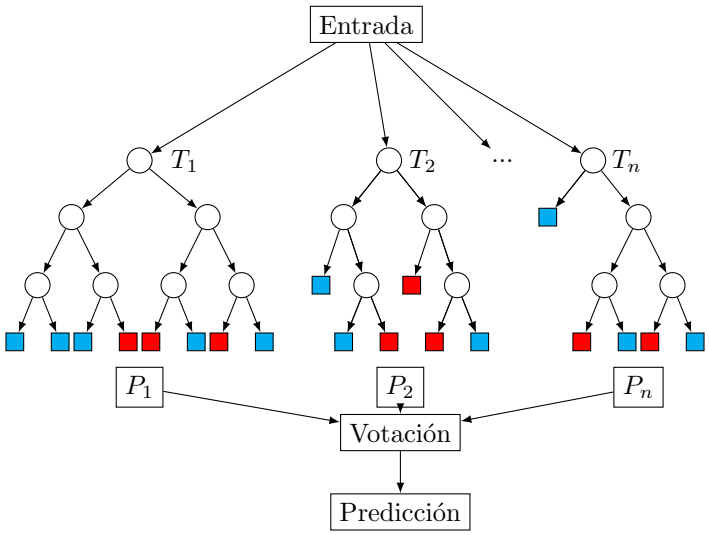


Figura 1.8: Ensamble de árboles o Bosque. Elaboración propia.

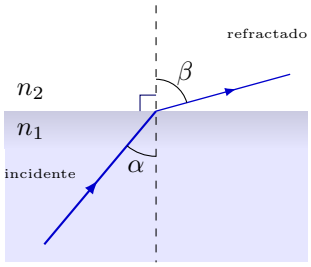


Figura 1.9: Refracción y reflexión de un haz de luz en una interfaz de dos medios con distintos índices de refracción. Elaboración propia.

1.4.2. Reflexión y refracción

Un rayo de luz incidente en una interfaz entre dos medios de diferente naturaleza cambiará de dirección. De acuerdo al ángulo de incidencia α_i, este cambio puede resultar en una refracción o en una reflexión total interna.

Considerando el ángulo de incidencia α_i y los medios n_1 y n_2, de distinto comportamiento óptico. Un rayo de luz estará sujeto a refracción de acuerdo a la ley de Snell.

$$n_1 \cdot sin(\alpha_i) = n_2 \cdot sin(\beta) \tag{1.16}$$

En la Ecuación (1.16), n_1y n_2 son índices de refracción de sus respectivos medios re-

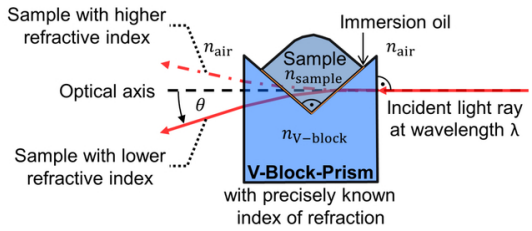


Figura 1.10: Diagrama de un V-Block, que ilustra el principio de medición de un refractómetro. Tomado de Petzold (2018).

fracción. Una forma de medir el índice de refracción de una muestra (Petzold, 2018) es utilizando un dispositivo conocido como V-Block, ilustrado en la Figura 1.10. El V-Block es utilizado para sostener y orientar un objeto óptico, como un prisma, un espejo o un cristal, en un ángulo preciso de 45 grados, el cual permite una precisa medición del índice de refracción.

1.4.3. Composición química e índice de refracción

El índice de refracción es una medida de la velocidad a la que la luz se mueve a través de un material, y esta velocidad se ve afectada por la naturaleza de cada material. Por ejemplo, los materiales con enlaces químicos más fuertes, como los materiales cerámicos, suelen tener un índice de refracción más alto que los materiales orgánicos, cuyos enlaces químicos son más débiles.

1.4.4. Birrefringencia

El término birrefringencia se refiere a la propiedad de algunos materiales de tener índices de refracción diferentes para diferentes polarizaciones de la luz que se propagan a través de ellos. Esto se debe a que las propiedades físicas de los materiales anisotrópicos, como el índice de refracción, varían en diferentes direcciones ya sea por su estructura cristalina o molecular.

1.4.5. Ecuación de Sellmeier

La ecuación de Sellmeier es una fórmula empírica que describe la relación entre el índice de refracción de un material y la longitud de onda de la luz que se está propagando a través de ese material. Está dada por la expresión de la Ecuación (1.17)

$$n^2(\lambda) = A + \frac{B_1\lambda^2}{\lambda^2 - C_1} + \frac{B_2\lambda^2}{\lambda^2 - C_2} + \frac{B_3\lambda^2}{\lambda^2 - C_3}. \tag{1.17}$$

La ecuación de Sellmeier es una elección conveniente en regiones donde no hay absorción debido a su simplicidad y su capacidad para describir con precisión la dispersión óptica de materiales transparentes. Aunque no tiene en cuenta los efectos de absorción, puede proporcionar una descripción adecuada de la refracción de la luz en un rango específico de longitudes de onda.

1.5. Conclusión del Capítulo

En este capítulo se presentó un resumen contextual que incluye las definiciones relevantes para inducir el modelo de aprendizaje automático de bosque aleatorio (Random Forest), usado en este trabajo. Se ha definido el aprendizaje automático como una rama de la inteligencia artificial que se enfoca en desarrollar algoritmos y modelos capaces de aprender y mejorar automáticamente a partir de datos. Finalmente se incluyó una breve explicación del fenómeno óptico producido por la interacción de la luz con la materia, usando la Ecuación de Sellmeier (Ecuación (1.17)) lo cual tiene relevancia en el contexto en el que se aplicará el modelo de aprendizaje automático desarrollado. A continuación, se presentará una propuesta metodológica para alcanzar los objetivos de identificación, usando todos los elementos definidos en este capítulo

Capítulo 2. Marco metodológico

En este capítulo se presenta la metodología propuesta para alcanzar el objetivo de este trabajo: identificar compuestos transparentes usando los datos de la respuesta óptica del material. Para la construcción del clasificador, se combinaron técnicas de reconocimiento óptico de caracteres (OCR), aumento de datos mediante modelos matemáticos, entrenamiento y cálculo de métricas de evaluación de clasificación, las cuales se detallan a lo largo del capítulo. En la Sección 2.1 se generaliza el problema de síntesis de datos para aprendizaje automatizado, para posteriormente en la Sección 2.1.1 presentar la metodología a usar en la construcción del clasificador, junto con sus limitaciones. En la Sección 2.2 se presenta la literatura científica desde donde se extraen las observaciones que darán pie a los datos del conjunto de entrenamiento. La extracción y estructuración de datos son presentadas en la Secciones 2.3 y 2.4. La Sección 2.6 muestra una técnica para abordar el desbalance en la cantidad de observaciones en el conjunto de entrenamiento, usando como criterio agrupación en regiones del espectro de luz. La Secciones 2.7 y 2.8 detallan las técnicas que se usaron para ajuste y evaluación del modelo de clasificación.

2.1. Identificación de la problemática

El principal desafío es encontrar la manera más efectiva de generar y utilizar los datos de entrenamiento. En el caso del fenómeno óptico de interés de este trabajo, la cantidad de datos disponibles no siempre alcanza un mínimo para inducir un clasificador, por lo que una generación sintética es necesaria. Para ello, se utilizan las técnicas de (Tatian, 1984), en dónde se muestra un método para ajustar los parámetros de la ecuación de Sellmeier (ver Ecuación (1.17)), con datos de índice de refracción de mediciones de longitud de onda, en regiones sin absorción del material medido (materiales transparentes para el caso de la luz visible).

2.1.1. Metodología propuesta

Para alcanzar los objetivos generales y específicos, la metodología propuesta se basa en el trabajo presentado en el artículo publicado por (Bikku *et al.*, 2022). Para este trabajo es la siguiente:

- Extracción de datos de índice de refracción desde literatura.
- Interpolación de curvas de Sellmeier para compuestos en regiones visible del espectro luminoso.
- Estructuración de datos y elaboración de un conjunto de entrenamiento.
- Muestreo de datos para incrementar representatividad de las regiones visibles.
- Ajuste del modelo clasificador Random Forest.
- Evaluación de la clasificación.

2.1.2. Limitaciones

A continuación se listan limitaciones identificadas con la metodología propuesta:

- Basa su soporte en la calidad del ajuste de la ecuación de Sellmeier.
- Los errores en los datos originales no son detectados con esta metodología: se asume que los datos extraídos desde la literatura son representativos para cada material.
- La metodología no propone una estructuración automatizada del conjunto de entrenamiento, por lo que se espera que la construcción de éste se realice de forma manual.

2.2. Orígenes de datos

Debido a que el fenómeno óptico en estudio puede ser descrito con la ecuación de Sellmeier específicamente en la región visible, los datos que se seleccionen para inducir un clasificador deben corresponder a mediciones realizadas en dicha región. Para este trabajo, el texto seleccionado fue *Nonlinear Optics of Organic Molecules and Polymers*

(Nalwa & Miyata, 1996), en particular el Capítulo 4, que contiene información óptica de materiales transparentes en la región visible. Dicho capítulo consta de 262 páginas con diversa información óptica, entre la cual se incluyen mediciones experimentales de índices de refracción, para diversas longitudes de onda (dentro de la región visible). Con esta información, en un documento PDF, se procederá a la elaboración de un conjunto de entrenamiento.

2.3. Extracción de datos

La extracción de datos se refiere al proceso de identificar y capturar información específica de un conjunto de datos, lo que puede implicar la transformación de información no estructurada en una estructura de datos más definida. A continuación se presentará la técnica de extracción de datos por reconocimiento óptico de caracteres, utilizada en este trabajo para obtener datos no estructurados de la literatura científica mencionada.

2.3.1. Extracción por OCR para obtención de datos tabulados

El reconocimiento óptico de caracteres (OCR) es una técnica utilizada para la extracción automatizada de datos de imágenes de texto impreso. En los últimos años, se ha producido un progreso significativo en la aplicación del aprendizaje automático a la inferencia de estructuras de tablas desde documentos. Esto debido a la creciente necesidad de extraer información estructurada y organizada de grandes cantidades de datos no estructurados, como artículos científicos, reportes de laboratorio, registros médicos, entre otros.
Para este trabajo, se aplicó un flujo de diferentes herramientas OCR las cuales generaron un conjunto de datos base para la construcción del clasificador. En una primera etapa se usó el software libre de OCR Tesseract (Smith, 2007) para identificar las páginas del texto que contenían las palabras “refractive index”. Posteriormente con aquellas páginas seleccionadas se utilizaron diferentes herramientas para identificar la posición de las tablas en la página, extraer los valores en ellas y generar un documento de texto separado por comas (csv). Entre las herramientas usadas para esta tarea se incluyen TabbyPDF (Shigarov *et al.*, 2018), PubTables-1M (Smock *et al.*, 2021) y Tabula (Aristarán *et al.*, 2012). La extracción de los índices de refracción a partir de imágenes de texto impreso permitió reconocer los caracteres escritos en diferentes fuentes, tamaños y estilos, sin embargo,

uno de los mayores desafíos en la inferencia de estructuras de tablas es la variabilidad en la estructura y el formato de las tablas en diferentes documentos. Si bien, los algoritmos de aprendizaje automático pueden adaptarse a esta variabilidad, las estructuras resultantes no siempre son las idóneas para la aplicación buscada (en este trabajo, la elaboración de un clasificador), por lo cual, una curación manual fue realizada sobre los datos de salida de estas herramientas.

2.4. Estructuración de datos

El objetivo de la estructuración de datos en esta etapa, es preparar los datos para que sean adecuados para su uso por el modelo clasificador. Los datos de entrada del conjunto de entrenamiento tienen una estructura específica (ver Ecuación (1.2)). En dicha estructura se incluyeron los valores de longitud de onda e índice de refracción, agregando como clase de esos valores, el nombre del compuesto asociado a esa observación. Cabe señalar que diversos materiales medidos en el texto original correspondían a materiales con dos o incluso tres índices de refracción, los cuales fueron etiquetados para poder realizar un aumento de dato sobre cada serie de puntos por separado.

2.5. Generación de datos usando la ecuación de Sellmeier

La idea de generar un conjunto de datos sintéticos es ampliar y mejorar la calidad del conjunto de datos original. Las técnicas descritas en (Tatian, 1984) fueron usadas como base para obtener una serie de datos aumentados a partir de mediciones específicas de longitud de onda.

La metodología consiste en tomar puntos del compuesto de la ecuación de Sellmeier (Ecuación (1.17)) usando un ajuste por mínimos cuadrados de los parámetros A, B_1, C_1, B_2, C_2. Solo se consideran tres términos de la ecuación, ya que no hay una reducción significativa del error al agregar más de tres términos (Tatian, 1984). En esta etapa también se descartó entregar soporte a aquellos compuestos que tuvieran menos de tres mediciones de índice de refracción. El ajuste fue realizado utilizando la librería `lmfit` en Python, la cual utiliza el algoritmo de Levenberg–Marquardt para un ajuste de expresiones no lineales mediante el error cuadrado medio (Nocedal & Wright, 1999).

Una vez ajustados los parámetros, se realizó una evaluación de cada expresión resultante

entre la menor y mayor longitud de onda. Los datos evaluados de cada serie fueron agrupados por compuestos, sin la etiqueta de su eje óptico y sin incluir los datos originales (observados) desde los cuales se obtuvieron.
Adicionalmente, los datos observados fueron puestos en un conjunto de validación, al cual se referirá como conjunto de datos observados, usado en la etapa de evaluación de clasificación.

2.6. Muestreo en conjuntos de entrenamiento desbalanceados

Al enfrentarse a un problema de clasificación, con frecuencia hay una diferencia entre el número de ejemplos de cada clase. Como el objetivo de este trabajo es elaborar un clasificador que pueda distinguir compuestos en la región visible del espectro, se busca dar mayor representatividad a esa región específica (datos que cumplan $0{,}4 < \lambda < 0{,}78$). Esto se consigue utilizando técnicas de muestreo para equilibrar el conjunto de entrenamiento y aumentar la representatividad en dicha región. La técnica utilizada para resolver el desbalance fue el sobremuestreo u *Oversampling* (Fernández *et al.*, 2018). Esta técnica consiste en aumentar el número de instancias de la clase minoritaria mediante la duplicación aleatoria de estas. La ventaja del *oversampling* es que no hay pérdida de información ya que todas las instancias de la clase minoritaria se conservan y solo se aumenta su representación en el conjunto de datos. Sin embargo el *oversampling* también puede conducir a un sobreajuste de los datos.
Otra alternativa para balancear el conjunto de entrenamiento es la técnica de submuestreo, la que consiste en eliminar algunas instancias de la clase mayoritaria para equilibrar el número de instancias entre ambos grupos. Sin embargo, se declinó utilizar esta técnica ya que puede significar una pérdida de información importante, por la posible eliminación de instancias relevantes de la clase mayoritaria.
El objetivo de ambas técnicas es obtener una representación equitativa de ambos grupos del conjunto de entrenamiento, para que sean igualmente representados al momento de realizar la inducción del clasificador.

2.7. Ajuste del modelo clasificador

Para entrenar un modelo clasificador Random Forest se utilizó el conjunto de entrenamiento etiquetado, obtenido posterior al aumento por ecuación de Sellmeier. El modelo fue implementado mediante la librería `scikit-learn` (Pedregosa *et al.*, 2011), con los hiperparámetros por defecto: se utiliza el criterio de impureza de Gini para evaluar la calidad de la división en cada nodo del árbol y el número máximo de características utilizadas en cada árbol se establece en la raíz cuadrada del número total de características. El número de árboles se estableció en 100 y no se aplicaron restricciones por altura o número de hojas.

Se separaron los datos en proporción 9:1, para entrenamiento y evaluación. La separación consiste en un muestreo aleatorio, sin repetición, desde el conjunto de entrenamiento, reservando un 10 % de los datos para pruebas. Además, se evaluó el modelo con el conjunto de validación, usando los datos observados que fueron obtenidos en la etapa de extracción por OCR. Ambos conjuntos de prueba fueron evaluados y reportados por separado, en distintas matrices de confusión.

2.8. Evaluación de la clasificación

A continuación se presentan las medidas de desempeño de clasificación usadas en este trabajo, las cuales van desde mediciones básicas de verdaderos y falsos positivos hasta medidas más complejas como la validación cruzada y el área bajo la curvas ROC. Estas medidas corresponden a técnicas reportadas en problemas de clasificación y diagnóstico.

2.8.1. Matriz de confusión

Se consideró en primera instancia el uso de la matriz de confusión (Swets, 1988), la cual ilustra gráficamente la cantidad de predicciones acertadas v/s erróneas de la clasificación de cada clase.

Además se computaron las medidas de las Ecuaciones (2.1) a (2.4):

- Exactitud (accuracy):

$$\frac{TP + TN}{TP + FN + TN + FP} \tag{2.1}$$

- Sensibilidad (recall o sensitivity):

$$TP_{rate} = \frac{TP}{TP + FN} \tag{2.2}$$

- Tasa Falsos Positivos:

$$FP_{rate} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{2.3}$$

- Precisión:

$$PP_{value} = \frac{TP}{TP + FP} \tag{2.4}$$

Donde TP, TN, FP, FN corresponden a Verdaderos Positivos, Verdaderos Negativos, Falsos Positivos y Falsos Negativos respectivamente.

2.8.2. Validación cruzada

La validación cruzada (Hastie *et al.*, 2009) es una técnica ampliamente usada para estimar el error esperado en muestras no vistas, es decir, la capacidad de generalización del modelo entrenado. En particular el método *K-fold* usa una parte (o pliegue) de los datos disponibles para ajustar el modelo y una parte diferente de éste para evaluarlo.
En este trabajo es utilizó un valor de $K = 10$, entrenando en cada caso con un 20 % de los datos, seleccionados al azar.

2.8.3. Área bajo la curva ROC

El área bajo la curva ROC (Fawcett, 2006) mide la capacidad del modelo para distinguir entre clases positivas y negativas. Está diseñado para clasificaciones binarias, por lo que se adaptó para este trabajo utilizando una metodología One-versus-Rest (OvR) (Bishop, 2006) para medir el rendimiento de cada clase. La metodología entrena un clasificador binario para cada clase, calcula la curva ROC para esa clase, finalmente calcula la media aritmética entre todas las clases.
La macro-media del área bajo la curva ROC proporciona una medida del rendimiento del modelo clasificador en un problema de clasificación múltiple. Esta medida de rendimiento de clasificación se interpreta como la capacidad del modelo para distinguir entre las clases, en términos de la tasa de verdaderos postivos y falsos positivos.
Un valor de ROC igual o inferior a 0.5 indica un rendimiento peor que una clasificación al azar, mientras que un valor de 1 indica una clasificación perfecta.

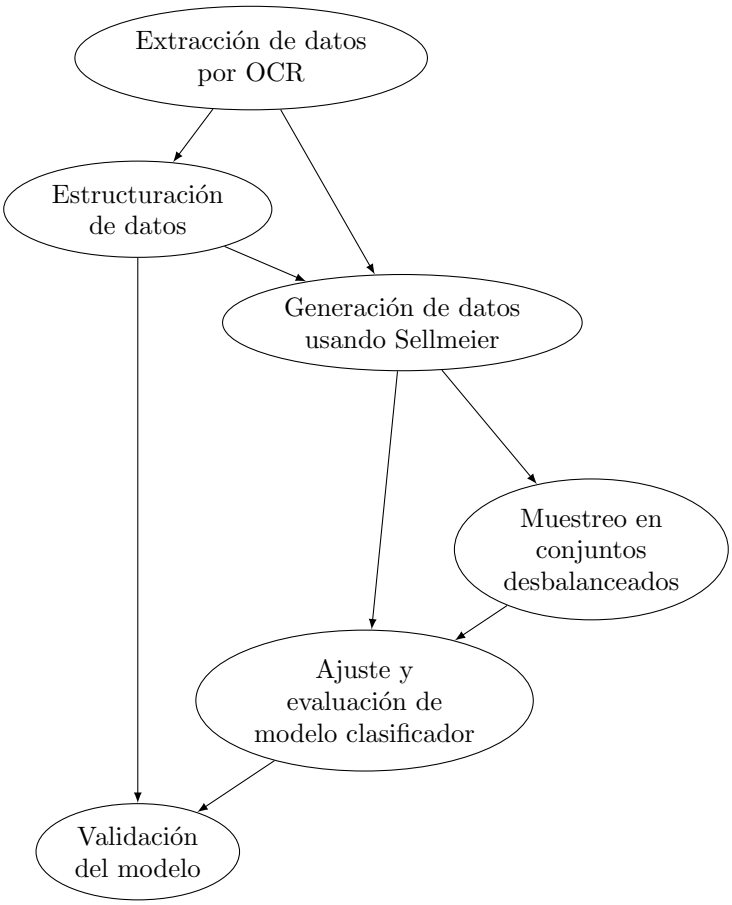


Figura 2.1: Metodología propuesta para identificar compuestos mediante su respuesta óptica. Elaboración propia.

2.9. Conclusión del capítulo

En este capítulo, se presentó la metodología propuesta para alcanzar el objetivo del trabajo. Se combinaron diferentes técnicas, como el OCR, el aumento de datos, el entrenamiento del modelo y la evaluación del clasificador. Además, se detallaron los pasos relacionados con la síntesis de datos, la extracción y estructuración de los mismos, la corrección del desbalance en el conjunto de entrenamiento y el ajuste y la evaluación del modelo. Estas técnicas fueron aplicadas bajo una secuencia lógica de actividades, como se ilustra en la Figura 2.1. A continuación se presentarán los resultados de la implementación de esta metodología

Capítulo 3. Resultados y análisis

En este capítulo, se presentan los resultados de los experimentos y se analizan en función de los objetivos planteados. En primera instancia se caracteriza la plataforma computacional utilizada (Sección 3.1). Los resultados se dividen en dos secciones, la primera sección corresponde a “Procesamiento y generación de datos” (Sección 3.2), la que incluye los resultados de la extracción por OCR, caracterización y generación de datos sintéticos. La sección de “Entrenamiento y evaluación” (Sección 3.3), muestra el rendimiento obtenido por el modelo clasificador, con y sin tratamiento de datos, con el conjunto de entrenamiento y validación.

3.1. Plataforma computacional

La plataforma computacional utilizada para el desarrollo del proyecto se basó en el siguiente hardware:

- Procesador: Intel Core i5 9400F, 6 núcleos (2.9-4.1 GHz).
- Memoria RAM: 32GB.

En cuanto al lenguaje de programación, se utilizó Python 3.8 para el desarrollo, en particular, se hizo uso de la biblioteca scikit-learn (Pedregosa *et al.*, 2011), por su diversidad de herramientas para la clasificación y preprocesamiento de datos.

3.2. Procesamiento y generación de datos

En esta sección se muestran los resultados del procesamiento y estructuración de datos, así como los resultados de la creación de datos sintéticos. Los resultados son presentados según las siguientes etapas:

1. Extracción por OCR.
2. Curación de datos y estructuración.
3. Generación de datos sintéticos usando la ecuación de Sellmeier.

3.2.1. Extracción de datos por OCR

En la primera etapa de reconocimiento, las páginas seleccionadas por la herramienta *Tesseract* (Smith, 2007) redujeron el documento PDF de 264 a 54 páginas, lo que facilitó la búsqueda por las subsiguientes herramientas de OCR. En cuanto a la identificación de tablas, la extracción automatizada entregó una serie de tablas, las cuales asociadas al número de página, fueron curadas manualmente para solo incluir y estructurar los datos de las observaciones de respuesta óptica para cada material (Figura 3.1).

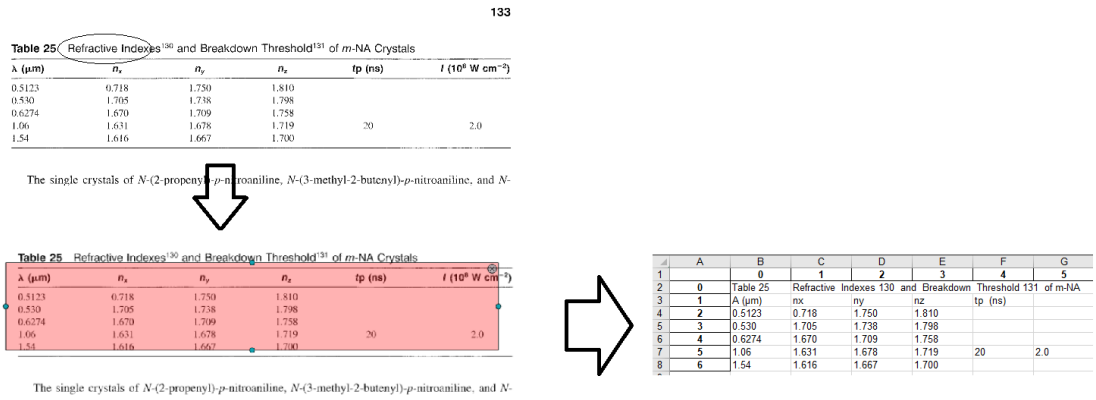


Figura 3.1: Extracción por dos etapas de OCR. La primera etapa filtró el contenido buscando “refractive index” (ilustrado en la parte superior izquierda), luego se extrajo automáticamente en datos semi-estructurados (lado inferior izquierdo y derecho de la imagen). Elaboración Propia.

Los datos de cada material, incluyendo sus distintos índices ópticos, fueron finalmente puestos en una tabla donde se procedió a las siguientes etapas de la construcción

del conjunto de entrenamiento. Una representación de la estructuración de los datos extraídos, mediante una distribución conjunta para λ y n es ilustrada en la Figura 3.2. La caracterización se encuentra en la Tabla 3.1, donde destaca la cantidad de 67 series de datos extraídos para cada índice óptico de cada material. Estas series son entregadas a la etapa de aumento de datos.

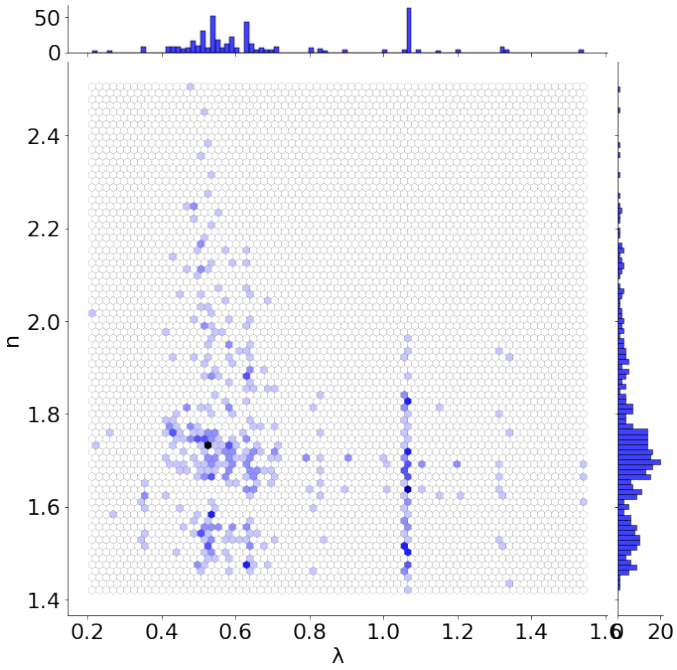


Figura 3.2: Distribución conjunta para λ y n, resultado de la extracción por OCR. Cada sección representa el número de observaciones reportadas en (Nalwa & Miyata, 1996), capturadas por las herramientas OCR. Elaboración Propia.

Tabla 3.1: Caracterización de columnas del conjunto de datos extraídos. Incluye los índices ópticos de todos los materiales, cuando aplica.

Columna	book	λ	n	k
conteo	408	408	408	408
tipo	categórica	numérica	numérica	numérica
únicos	67	-	-	-

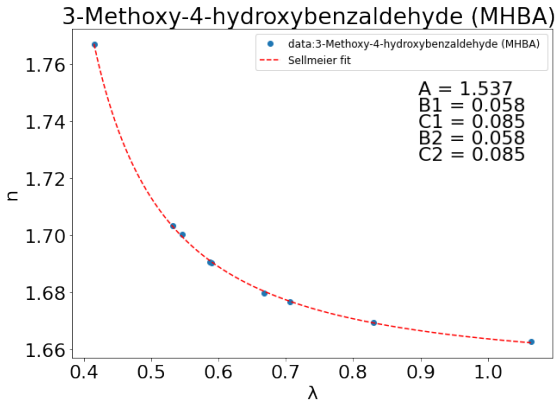


Figura 3.3: Ajuste e interpolación de datos para un compuesto (MHBA). Los parámetros A, B1, B2, C1, C2 corresponden a los valores de la ecuación de Sellmeier con 3 términos. Elaboración Propia.

3.2.2. Datos sintéticos usando ecuación de Sellmeier

Con los datos de cada compuesto se realizó un ajuste de tres términos de la ecuación de Sellmeier. La obtención de parámetros se realizó usando métodos de la librería `lmfit` (Newville *et al.*, 2022), con restricciones en los valores basadas en la metodología de (Tatian, 1984). Los parámetros obtenidos de cada compuesto fueron puestos en la Ecuación (1.17) y evaluados en el intervalo que contenía al valor máximo y mínimo de λ de los puntos de datos cada serie. El intervalo fue dividido en 3000 partes iguales, lo que generó una representación de la curva de Sellmeier de cada compuesto, como se puede ver ejemplificado en la Figura 3.3. El resultado del aumento de datos fue un conjunto de 201.000 observaciones sintéticas, con soporte a 25 materiales según su respuesta óptica. Los datos aumentados, que representan al conjunto de entrenamiento, están representados en la distribución de la Figura 3.4.

3.3. Entrenamiento y evaluación

En esta sección se presentan los resultados obtenidos con la metodología propuesta en la inducción del modelo Random Forest y la calidad de la clasificación entregada por este. Se detallan los dos tipos de entrenamientos del modelo con datos con y sin balance de etiquetas, mostrando las métricas de rendimiento para el conjunto de entrenamiento y el conjunto de validación. Para cada caso se registraron métricas de rendimiento permitiendo comparar la efectividad del tratamiento de datos y visualizar el soporte obtenido por cada

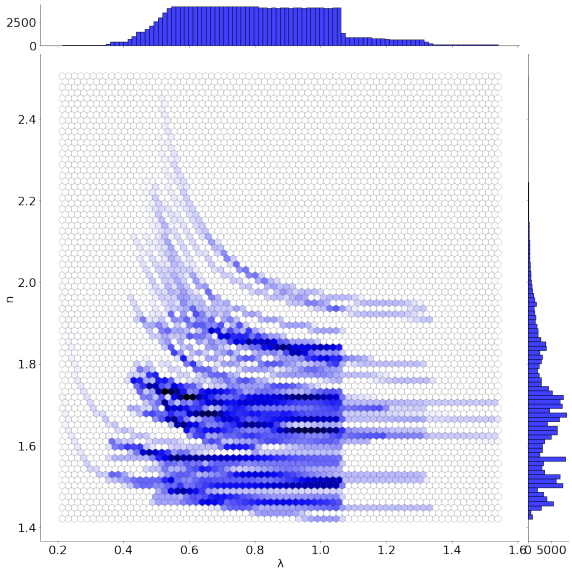


Figura 3.4: Distribución conjunta para λ y n, con datos aumentados por la ecuación de Sellmeier. Los datos ilustrados en la Figura corresponden al conjunto de datos utilizado para entrenamiento del clasificador. Elaboración Propia.

una de las clases del conjunto. Posteriormente se analizan otras métricas de rendimiento del clasificador como la validación cruzada y el área bajo la curva ROC.

3.3.1. Entrenamiento y rendimiento de la clasificación

Se realizó el ajuste del modelo Random Forest con hiper-parámetros obtenidos de (Breiman, 1996, 2001), con los datos aumentados por la ecuación de Sellmeier, obteniendo las métricas de rendimiento de la columna “augmented data” en la Tabla 3.2. El rendimiento con el conjunto de validación se reporta a continuación en la columna “observed set‘”. Estos resultados se utilizan como línea base para comparar el rendimiento del modelo entrenado con tratamiento de datos. El detalle del soporte de clasificación se incluye como matrices de confusión en el Apéndice B, en las Figuras B.1 a B.4. Este resultado contribuye directamente al cumplimiento del objetivo general planteado en el trabajo: identificar entre materiales transparentes usando la respuesta óptica en la región visible del espectro.

3.3.2. Muestreo en la región visible

Inicialmente, el número de muestras en la región visible fue de 95.499, el número de entradas fuera de dicha región corresponde a 105.501. El desequilibrio del número de

etiquetas en la región visible es alrededor del 3 % del total de las observaciones, por lo que se realizó un sobre-muestreo duplicando aleatoriamente entradas dentro de la región hasta igualar el número de entradas dentro y fuera del espectro visible. El resultado fue un incremento del tamaño del conjunto de entrenamiento de 201.000 a 211.002. Las distribuciones de observaciones antes y después del muestreo se ilustran en la Figuras 3.5 y 3.6. Posteriormente, se indujo un nuevo modelo Random Forest sobre el conjunto de datos balanceado, evaluando su rendimiento con ambos conjuntos de prueba y validación. Las matrices de confusión resultantes están en las Figuras B.5 y B.6.

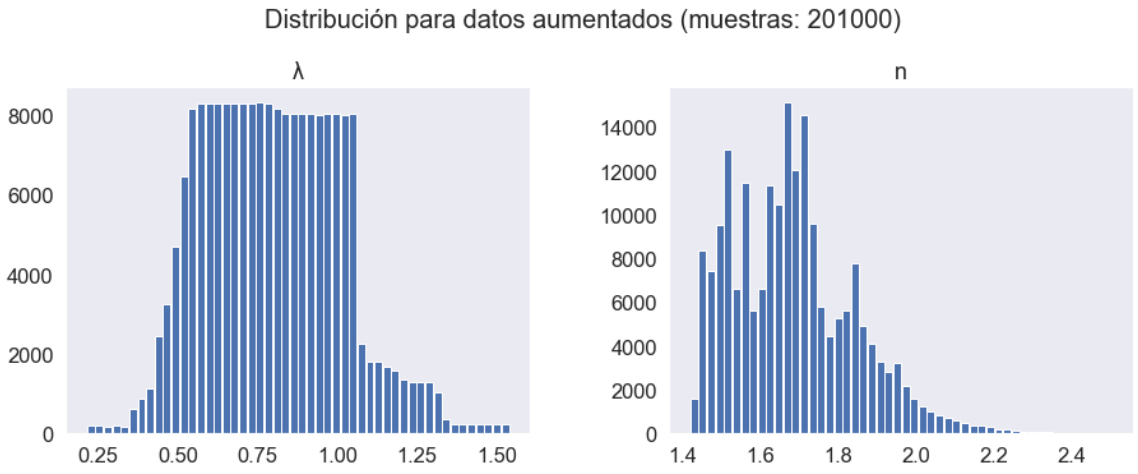


Figura 3.5: Distribución de datos previo al muestreo. Elaboración Propia.

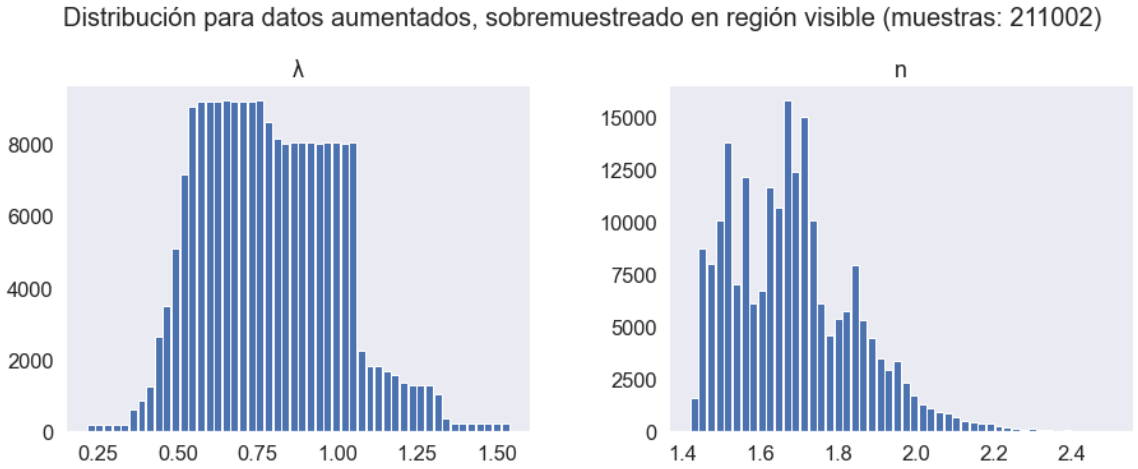


Figura 3.6: Distribución de datos, balanceados para muestras en la región visible. Elaboración Propia.

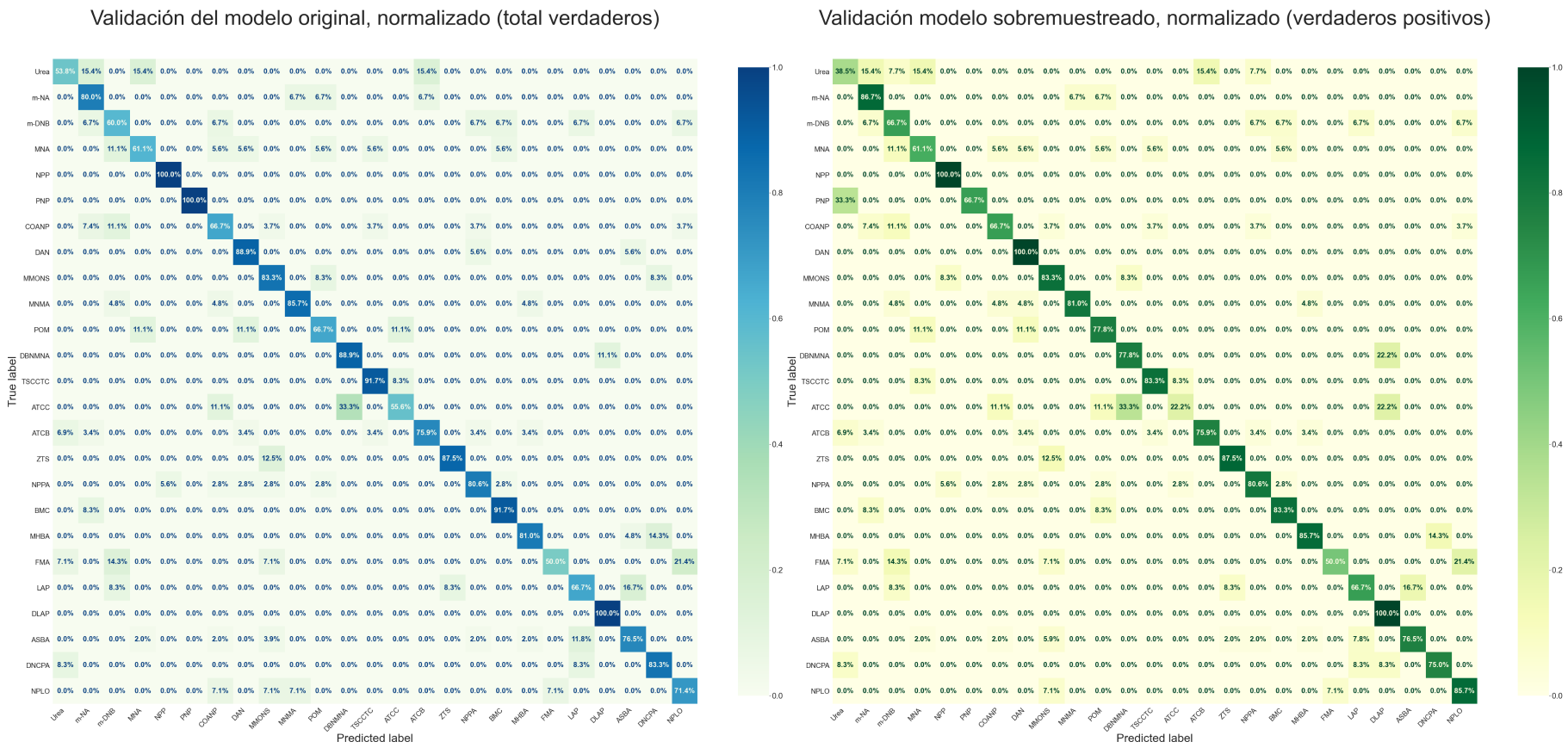


Figura 3.7: Las matrices de confusión muestran los resultados de validación de los modelo inducidos sobre datos aumentados y re-muestreados. Elaboración Propia.

Tabla 3.2: Resumen de métricas de desempeño

Training set:	augmented data trained model (original data)		resampled data trained model (balanced)	
Test set:	train-test set	observed set (validation)	train-test set	observed set
accuracy:	97.93 %	76.47 %	98.67 %	75.74 %
recall:	97.90 %	77.87 %	98.63 %	75.14 %
precision:	97.88 %	75.81 %	98.64 %	73.38 %
f1-score:	97.89 %	76.15 %	98.63 %	72.94 %

3.3.3. Métricas de desempeño

Se comparan ambas instancias del modelo Random Forest sobre los datos en la Tabla 3.2 con los valores de las métricas del rendimiento de clasificación para cada evaluación. Las columnas *augmented data* y *resampled data* corresponden al desempeño del clasificador sobre los datos sintéticos aumentados y con sobre-muestreo, respectivamente. En ambos casos, las columnas *observed set* corresponden al rendimiento de validación de los modelos con los datos originales de observaciones extraídas desde la literatura especializada (ver Tabla 3.1 y Figura 3.2).

Se observa en la Tabla 3.2 que el sobre-muestreo puede mejorar el rendimiento en las pruebas de entrenamiento, pero no se traduce en una mejora en la prueba de validación. Las matrices de confusión de la Figura 3.7 muestran en detalle el rendimiento de validación de la clasificación para cada material, antes y después del sobre-muestreo, observándose que hay tanto aumentos como disminuciones en el rendimiento de clasificación para cada clase. La disminución del rendimiento en el conjunto de validación sugiere que el modelo entrenado con datos sintéticos con tratamiento de datos puede estar “sobreajustado” y tener dificultades para generalizar sobre algunas clases y adaptarse a datos reales.

3.3.4. Validación cruzada

La validación cruzada se realizó usando el conjunto de datos desbalanceado (*augmented*), iterando el modelo RandomForest en 10 series, sobre el conjunto de entrenamiento particionando entre entrenamiento y prueba conforme a lo descrito en (Breiman, 2001) para datos sintéticos, usando cada vez el 80 % de los datos para medir la exactitud del

Tabla 3.3: Exactitud usando $k = 10$ en método k-fold

Fold	1	2	3	4	5
Exactitud	95.30 %	95.28 %	95.44 %	95.40 %	95.16 %
Fold	6	7	8	9	10
Exactitud	95.30 %	95.33 %	95.40 %	95.38 %	95.44 %

modelo entrenado con el 20 % de los datos. La selección de datos para cada iteración fue aleatoria, usando el método *ShuffleSplit* de *sklearn* (Pedregosa *et al.*, 2011). Los resultados de las 10 iteraciones se presentan en la Tabla 3.3, junto con una representación gráfica de los índices utilizados para entrenamiento y evaluación en la Figura 3.8.
La media de la exactitud después de la validación 10-Fold fue de 95.34 % ($\pm$ 0.16 %). Esto muestra que el modelo tiene capacidad de generalizar sobre los datos de entrenamiento, alcanzando un 95 % de exactitud con solo "ver" el 20 % de las observaciones.

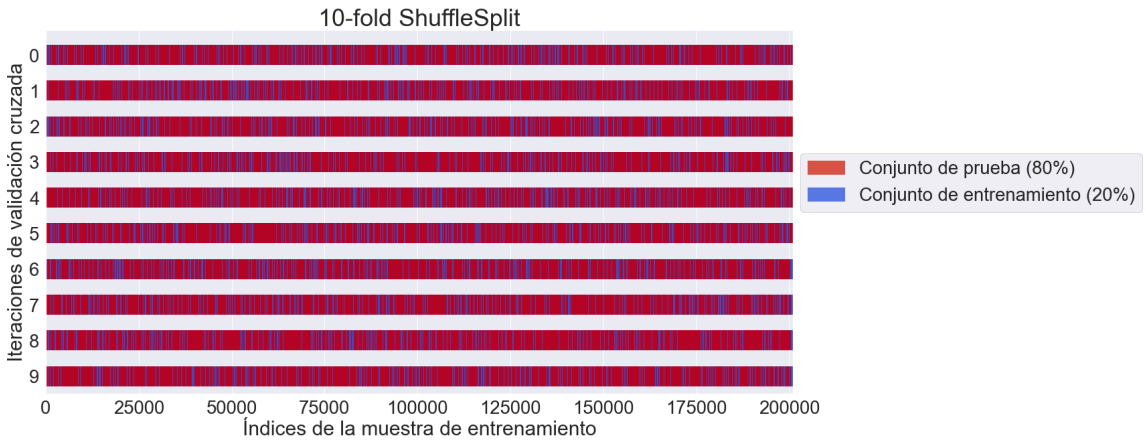


Figura 3.8: Partición aleatoria del los datos de entrenamiento, para cada pliegue del método *k-fold*. Elaboración Propia.

3.3.5. Área bajo la Curva ROC

La Área bajo la Curva ROC (AUC-ROC) se calculó a partir de las tasas de verdaderos vs falsos positivos, definido en las Ecuaciones (2.2) y (2.3), usando una metodología One-versus-Rest. La curva ROC representa la relación entre TPR y FPR y se utiliza para evaluar la capacidad del modelo de distinguir correctamente una clase, evaluando cuánto se equivoca en proporción a cuántas veces se predice dicha clase. El resultado de la

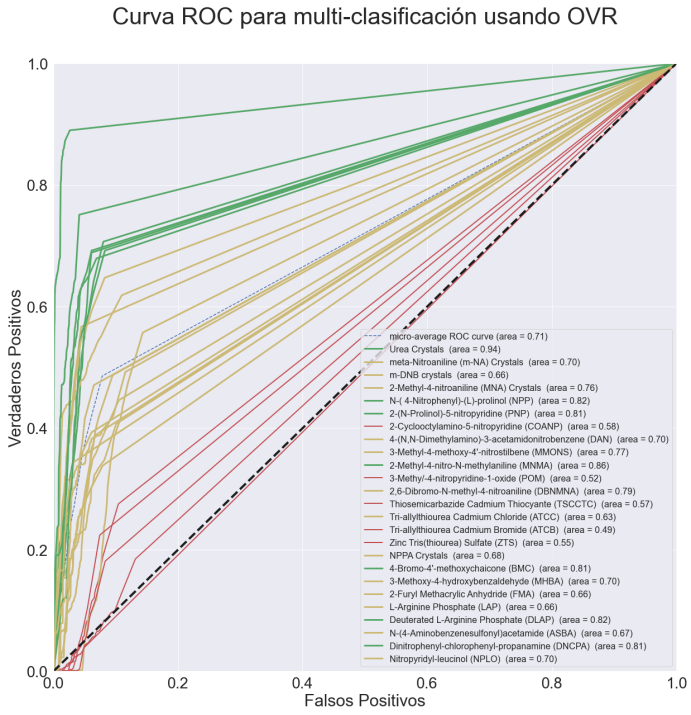


Figura 3.9: ROC para cada clase, usando estrategia OvR. Los colores indican la calidad de la clasificación para esa clase, siendo roja la peor calidad de clasificación con AUC $>$ 0.6, amarilla 0.6 $\geq$ AUC $<$ 0.8 y verde AUC $\geq$ 0.8. Elaboración Propia.

curva ROC para cada clase, con su respectiva área bajo la curva puede ser visto en la Figura 3.9. Computando el área bajo la curva se obtiene un indicador entre 0 y 1, para cada clase, asociado con la calidad del soporte para esa clase. Se usa el indicador AUC-ROC para establecer un criterio de calidad de la clasificación para cada compuesto, así, se agruparon los compuestos según su calidad de clasificación en tres grupos: Alto valor AUC ($>$0.8), valor intermedio ($\geq$ 0.6) y valor bajo ($<$0.6). La lista de materiales y la calidad de su clasificación están reportados en las Tablas 3.4 a 3.6.

Tabla 3.4: Materiales con bajo valor AUC.

Clase (material)	AUC
2-Cyclooctylamino-5-nitropyridine (COANP)	0.58
3-Methy/-4-nitropyridine-1-oxide (POM)	0.52
Thiosemicarbazide Cadmium Thiocyante (TSCCTC)	0.57
Tri-allylthiourea Cadmium Bromide (ATCB)	0.49
Zinc Tris(thiourea) Sulfate (ZTS)	0.55

Tabla 3.5: Materiales con valor AUC intermedio.

Clase (material)	AUC
meta-Nitroaniline (m-NA) Crystals	0.7
m-DNB crystals	0.66
2-Methyl-4-nitroaniline (MNA) Crystals	0.76
4-(N,N-Dimethylamino)-3-acetamidonitrobenzene (DAN)	0.7
3-Methyl-4-methoxy-4'-nitrostilbene (MMONS)	0.77
2,6-Dibromo-N-methyl-4-nitroaniline (DBNMNA)	0.79
Tri-allylthiourea Cadmium Chloride (ATCC)	0.63
NPPA Crystals	0.68
3-Methoxy-4-hydroxybenzaldehyde (MHBA)	0.7
2-Furyl Methacrylic Anhydride (FMA)	0.66
L-Arginine Phosphate (LAP)	0.66
N-(4-Aminobenzenesulfonyl)acetamide (ASBA)	0.67
Nitropyridyl-leucinol (NPLO)	0.7

Tabla 3.6: Materiales con valor AUC $\geq$ 0.8.

Clase (material)	AUC
Urea Crystals	0.94
N-(4-Nitrophenyl)-(L)-prolinol (NPP)	0.82
2-(N-Prolinol)-5-nitropyridine (PNP)	0.81
2-Methyl-4-nitro-N-methylaniline (MNMA)	0.86
4-Bromo-4'-methoxychaicone (BMC)	0.81
Deuterated L-Arginine Phosphate (DLAP)	0.82
Dinitrophenyl-chlorophenyl-propanamine (DNCPA)	0.81

3.3.6. Comparación con otros modelos clasificadores

Con los resultados de rendimiento, se llevó a cabo una comparación del modelo clasificador desarrollado con otros modelos clasificadores existentes. Se llevó a cabo por separado utilizando el conjunto datos aumentado desbalanceado y el conjunto presentado en (Bikku

et al., 2022), siguiendo una metodología estándar de validación cruzada “10-Fold ”. Los resultados de la comparación están disponibles en el Apéndice A.

3.4. Conclusión del capítulo

En este capítulo se caracterizó la plataforma computacional utilizada entregando información relevante acerca del contexto de hardware en el que se realizaron los entrenamientos al modelo “Random Forest”. En la Sección 3.2 se mostraron y caracterizaron los datos extraídos desde la literatura especializada usando técnicas de OCR. Se generaron datos sintéticos a partir de los datos extraídos, los cuales fueron entregados a la etapa de Entrenamiento y evaluación (Sección 3.3). Se realizaron ajustes al modelo con y sin tratamiento de datos, visualizando el impacto del muestreo en el rendimiento del modelo y su desempeño en validación. Se observa que, en general, el modelo alcanza un buen rendimiento, pero se destaca una disminución en el rendimiento cuando se aplica el tratamiento de datos en el conjunto de validación.

Capítulo 4. Conclusiones

En conclusión, en este trabajo se realizó la construcción de un modelo clasificador Random Forest, implementado con hiper-parámetros por defecto. La obtención de los datos de observaciones fue realizada de manera semi-automatizada usando herramientas de reconocimiento óptico de caracteres (OCR) y la confección del conjunto de datos utilizó técnicas de aumento de datos de observaciones, basadas en la teoría de dispersión de la luz en regiones transparentes del espectro luminoso, usando la ecuación de Sellmeier.

En base a los resultados obtenidos con respecto al balance del conjunto de entrenamiento, las técnicas de sobre-muestreo, incrementaron con éxito las precisiones ya que esta estrategia aumenta la representatividad de áreas menos exploradas por el modelo, sin embargo las mejoras no se reflejan en la validación del modelo, llegando incluso a reducir el desempeño en todas las métricas al probar el modelo entrenado con tratamiento de datos (ver Tabla 3.2).

De manera similar al trabajo presentado en el artículo de (Bikku *et al.*, 2022), se muestra que en una región transparente es posible realizar una identificación óptica de materiales, usando solo la información disponible en esa región del espectro. La principal diferencia entre ambos trabajos, es el uso exclusivo de datos de materiales sin coeficiente de absorción en todas las observaciones entregadas al modelo. Un trabajo futuro podría evaluar métodos de optimización de los hiper-parámetros del modelo clasificador, validando su capacidad de generalización. También se podría considerar incluir ruido aditivo en los valores obtenidos en la evaluación del ajuste de la ecuación de Sellmeier.

En la Figura 3.7, la matriz de confusión funciona como herramienta para identificar qué clases obtienen mayor tasa de error, posteriormente se utilizan los datos de clasificación para calcular un indicador de la calidad de la clasificación (Tablas 3.4 a 3.6). Identificar una correlación entre la tasa de error de cada clase con la calidad del ajuste a la ecuación de Sellmeier es un posible paso a seguir para estimar la capacidad de generalización del

modelo Random Forest.

Debido a las limitaciones en los orígenes de datos (disponibilidad de características y ajuste con posibles datos erróneos), evaluar estrategias desde el conocimiento del dominio del problema para aumentar la información del fenómeno óptico, podría también aumentar la el rendimiento del clasificador. Otro alternativa de validación puede ser integrar el modelo en un dispositivo de refractometría, que permita realizar pruebas reales de clasificación, determinando así la capacidad de generalización del modelo propuesto en este trabajo.

Referencias bibliográficas

Aristarán, M., Tigas, M., & Merrill, J. B. (2012). Tabula: Extract Tables from PDFs. https://tabula.technology/

Bikku, T., Fritz, R. A., Colón, Y. J., & Herrera, F. (2022). Machine learning identification of organic compounds using visible light. https://doi.org/10.48550/arxiv.2204.11832

Bikku, T., & Herrera, F. (2022). fherreralab/organic_optical_classifier: v1.0.0. https://doi.org/10.5281/ZENODO.6419971

Bishop, C. M. (2006). *Pattern Recogniton and Machine Learning*. Springer-Verlag New York.

Breiman, L. (1996). Bagging predictors. *Machine Learning 1996 24:2*, *24*(2), 123-140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1). https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification And Regression Trees*. Routledge. https://doi.org/10.1201/9781315139470

Buchner, F., Wasem, J., & Schillo, S. (2017). Regression Trees Identify Relevant Interactions: Can This Improve the Predictive Performance of Risk Adjustment? *Health Economics*, *26*(1), 74-85. https://doi.org/10.1002/HEC.3277

Cover, T. M., & Thomas, J. A. (2005). *Elements of Information Theory* (2nd ed.). Wiley. https://doi.org/10.1002/047174882X

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861-874. https://doi.org/10.1016/J.PATREC.2005.10.010

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from Imbalanced Data Sets*. Springer International Publishing. https://doi.org/10.1007/978-3-319-98074-4

Gründler, P. (2007). *Chemical sensors: An introduction for scientists and engineers*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-45743-5

Hakim, M. A., Jahan, N., Zerin, Z. A., & Farha, A. B. (2021). Performance Evaluation and Comparison of Ensemble Based Bagging and Boosting Machine Learning Methods for Automated Early Prediction of Myocardial Infarction. *2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021*. https://doi.org/10.1109/ICCCNT51525.2021.9580063

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Elements of Statistical Learning* (2nd ed., Vol. 27).

Hyafil, L., & Rivest, R. L. (1976). Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, *5*(1), 15-17. https://doi.org/10.1016/0020-0190(76)90095-8

Kubat, M. (2017). *An Introduction to Machine Learning*. Springer International Publishing. https://doi.org/10.1007/978-3-319-63913-0

Levitin, A. (2012). *Introduction to the design & analysis of algorithms* (3rd ed.). Pearson Education.

Madden, M. G., & Howley, T. (2009). A Machine Learning Application for Classification of Chemical Spectra. En *Applications and Innovations in Intelligent Systems XVI* (pp. 77-90). Springer London. https://doi.org/10.1007/978-1-84882-215-3_6

Madden, M. G., & Ryder, A. G. (2003). Machine learning methods for quantitative analysis of Raman spectroscopy data. En T. J. Glynn (Ed.), *https://doi.org/10.1117/12.464039* (p. 1130). SPIE. https://doi.org/10.1117/12.464039

Mohan, S., Kato, E., Drennen, J. K., & Anderson, C. A. (2019). Refractive Index Measurement of Pharmaceutical Solids: A Review of Measurement Methods and Pharmaceutical Applications. *Journal of pharmaceutical sciences*, *108*(11), 3478-3495. https://doi.org/10.1016/J.XPHS.2019.06.029

More, A. S., & Rana, D. P. (2017). Review of random forest classification techniques to resolve data imbalance. *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, 72-78. https://doi.org/10.1109/ICISIM.2017.8122151

Morgan, J. N., & Sonquist, J. A. (1963). Problems in the Analysis of Survey Data, and a Proposal. *Journal of the American Statistical Association*, *58*(302), 415. https://doi.org/10.2307/2283276

Nalwa, H. S., & Miyata, S. (Eds.). (1996). *Nonlinear Optics of Organic Molecules and Polymers* (1st Edition). CRC Press. https://doi.org/10.1201/9780138745493

Newville, M., Otten, R., Nelson, A., Stensitzki, T., Ingargiola, A., Allan, D., Fox, A., Carter, F., Michał, Osborn, R., Pustakhod, D., Ineuhaus, Weigand, S., Aristov, A., Glenn, Deil, C., Mark, Hansen, A. L. R., Pasquevich, G., … Hahn, A. (2022). lmfit/lmfit-py: 1.1.0. https://doi.org/10.5281/ZENODO.7370358

Nocedal, J., & Wright, S. J. (1999). Nonlinear Least-Squares Problems. En J. Nocedal & S. J. Wright (Eds.), *Numerical Optimization* (pp. 250-275). Springer-Verlag. https://doi.org/10.1007/0-387-22742-3_10

Park, H., & Son, J. H. (2021). Machine Learning Techniques for THz Imaging and Time-Domain Spectroscopy. *Sensors 2021, Vol. 21, Page 1186*, *21*(4), 1186. https://doi.org/10.3390/S21041186

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830. http://jmlr.org/papers/v12/pedregosa11a.html

Pes, B. (2020). Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains. *Neural Computing and Applications*, *32*, 5951-5973. https://doi.org/10.1007/S00521-019-04082-3/TABLES/5

Petzold, U. (2018). Optical Glass: A High-Tech Base Material as Key Enabler for Photonics. *Advances in Glass Science and Technology*. https://doi.org/10.5772/INTECHOPEN.73925

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*(1), 81-106. https://doi.org/10.1023/A:1022643204877

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc. https://doi.org/https://dl.acm.org/doi/book/10.5555/152181

Ratner, D., Sumpter, B., Alexander, F., Billings, J. J., Coffee, R., Cousineau, S., Denes, P., Doucet, M., Foster, I., Hexemer, A., Hidas, D., Huang, X., Kalinin, S., Kiran,

M., Kusne, A. G., Mehta, A., Ramirez-Cuesta, A., Sankaranarayanan, S., Scott, M., ... Yager, K. (2019). [Office of Basic Energy Sciences (BES)] Roundtable on Producing and Managing Large Scientific Data with Artificial Intelligence and Machine Learning. https://doi.org/10.2172/1630823

scikit-learn 1.1.2 documentation. (s.f.). Plot the decision surface of decision trees trained on the iris dataset. https://scikit-learn.org/stable/auto_examples/tree/plot_iris_dtc.html

Shigarov, A., Altaev, A., Mikhailov, A., Paramonov, V., & Cherkashin, E. (2018). TabbyPDF: Web-Based System for PDF Table Extraction. En *Communications in Computer and Information Science* (pp. 257-269). Springer Verlag. https://doi.org/10.1007/978-3-319-99972-2_20

Smith, R. (2007). An overview of the tesseract OCR engine. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, *2*, 629-633. https://doi.org/10.1109/ICDAR.2007.4376991

Smock, B., Pesala, R., & Abraham, R. (2021). PubTables-1M: Towards comprehensive table extraction from unstructured documents, 4624-4632. https://doi.org/10.48550/arxiv.2110.00061

Steen, W. M., & Mazumder, J. (2010). *Laser material processing: Fourth edition*. https://doi.org/10.1007/978-1-84996-062-5

Swets, J. A. (1988). Measuring the Accuracy of Diagnostic Systems. *Science*, *240*(4857), 1285-1293. http://www.jstor.org/stable/1701052

Tatian, B. (1984). Fitting refractive-index data with the Sellmeier dispersion formula. *Applied Optics, Vol. 23, Issue 24, pp. 4477-4485*, *23*(24), 4477-4485. https://doi.org/10.1364/AO.23.004477

Therneau, T., Atkinson, B., & Ripley, B. (2015). rpart: Recursive partitioning and regression trees. *R package version*, *4*, 1-9. https://cran.r-project.org/package=rpart

Way, M. J., Scargle, J. D., Ali, K. M., & Srivastava, A. N. (2012). Ensemble Methods: A Review. En *Advances in Machine Learning and Data Mining for Astronomy* (pp. 1-689). CRC Press. https://doi.org/10.1201/B11822

Witten, I. H., Frank, E., & Hall, M. A. (2011). Data mining : practical machine learning tools and techniques, 629.

Apéndice A. Comparación de modelos clasificadores

Los datos utilizados en la comparación de los modelos de clasificación provienen de dos orígenes, el conjunto A es el resultado de la metodología expuesta en el Capítulo 2 de esta tesis, mientras que el conjunto B corresponde al conjunto de datos obtenidos de (Bikku & Herrera, 2022). Los datos en estos conjuntos representan propiedades ópticas de los materiales, incluyendo longitud de onda, índice de refracción y coeficiente de absorción. Estas propiedades se representan en ambos conjuntos mediante el vector de la expresión de la Ecuación (A.1).

$$X = (\lambda, n, k) \tag{A.1}$$

El tamaño del conjunto A comprende un total de 201,000 entradas, con 3.000, 6.000 o 9.000 observaciones, dependiendo de la cantidad de índices ópticos de cada material. Por otro lado, el conjunto B consta de 452.069 entradas, con diversas cantidades de observaciones para cada clase. Para realizar la clasificación, se utilizó la herramienta Classification Learner de MATLAB para comparar el rendimiento del modelo clasificador desarrollado en esta tesis con otros modelos disponibles en la biblioteca de aprendizaje automático de MATLAB.

En cuanto a la división de los datos para la validación del modelo, en ambos casos, se reservó un 10 % de los datos como conjunto de validación y se realizaron 10 pliegues sobre los datos restantes. En la Tabla A.1, KNN obtuvo una alta exactitud en ambas pruebas (97.46 % y 97.48 %), quedando solo detrás de “Random Forest”. El modelo de Ensamble Random Forest muestra la más alta exactitud en ambos conjuntos. El modelo de Neural Network muestra una exactitud relativamente baja en ambos conjuntos, lo que indica que la arquitectura de la red neuronal debe optimizar sus hiper-parámetros o arquitectura para capturar las relaciones subyacentes en los datos de propiedades ópticas de los materiales.

La Tabla A.2 muestra el mismo patrón donde el modelo KNN y Random Forest, tienen un rendimiento significativamente mejor en términos de exactitud. El objetivo de esta comparación fue determinar el rendimiento del modelo clasificador desarrollado en relación con otros modelos existentes, hallando que KNN también es efectivo para clasificar materiales en función de sus propiedades ópticas.

Tabla A.1: Desempeño de clasificación en el Conjunto A

Tipo de Modelo	Exactitud % (10-fold)	Exactitud % (Validación)	Tiempo de entrenamiento (seg)
Árbol	47.67	47.57	17.556
KNN	97.46	97.42	13.236
Ensamble (Random Forest)	97.85	97.93	411.37
Red neuronal	27.38	32.37	355.41

Tabla A.2: Desempeño de clasificación en el Conjunto B

Tipo de Modelo	Exactitud % (10-fold)	Exactitud % (Validación B)	Tiempo de entrenamiento (s)
Árbol	62.63	62.81	32.225
KNN	91.36	91.13	25.12
Ensamble (Random Forest)	92.03	91.95	786.43
Red neuronal	53.85	55.87	1129.4

Apéndice B. Matrices de confusión de entrenamientos

Se presentan las matrices de confusión correspondientes a los resultados de la clasificación obtenidos en el trabajo, para los entrenamientos con y sin tratamiento de datos. Para ambos casos se incluye la evaluación hecha con el conjunto de validación.

Las matrices de confusión presentadas en este apéndice son una herramienta complementaria para respaldar los rendimientos reportados en el Capítulo 3 y las conclusiones presentadas en el Capítulo 4. Proporcionan una visión más detallada del desempeño del modelo clasificador en cada clase y permiten una evaluación más profunda de su capacidad predictiva.

Figura B.1: Matriz de confusión para datos aumentados. Elaboración Propia.

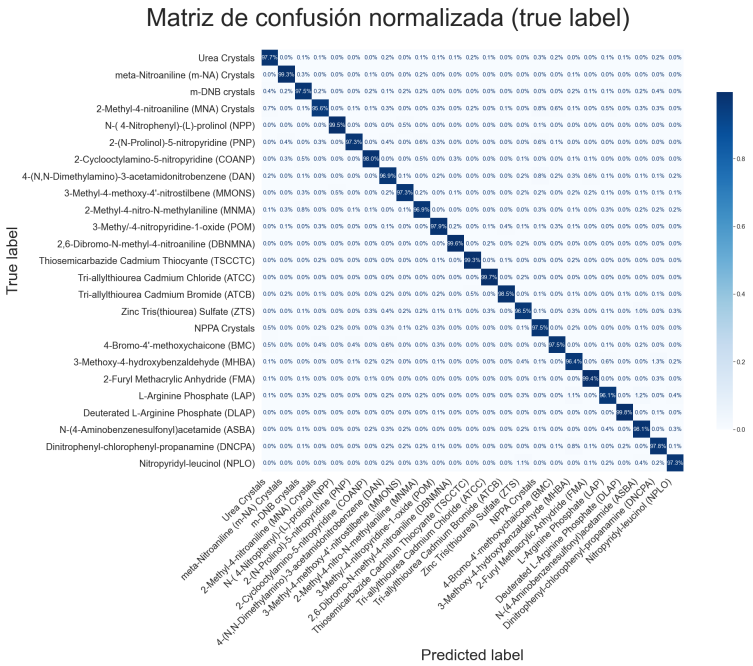


Figura B.2: Matriz de confusión para datos aumentados, normalizado. Elaboración Propia.

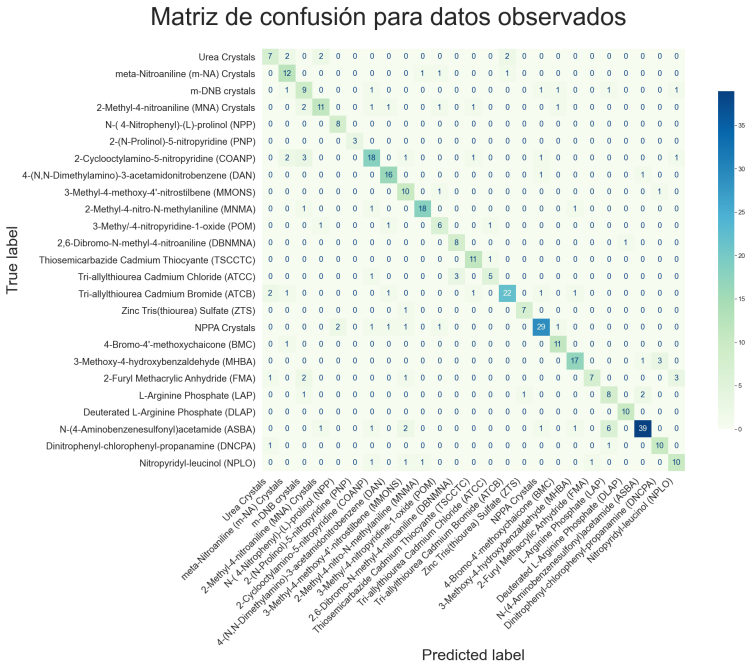


Figura B.3: Modelo entrenado con datos aumentados y evaluado con datos observados. Elaboración Propia.

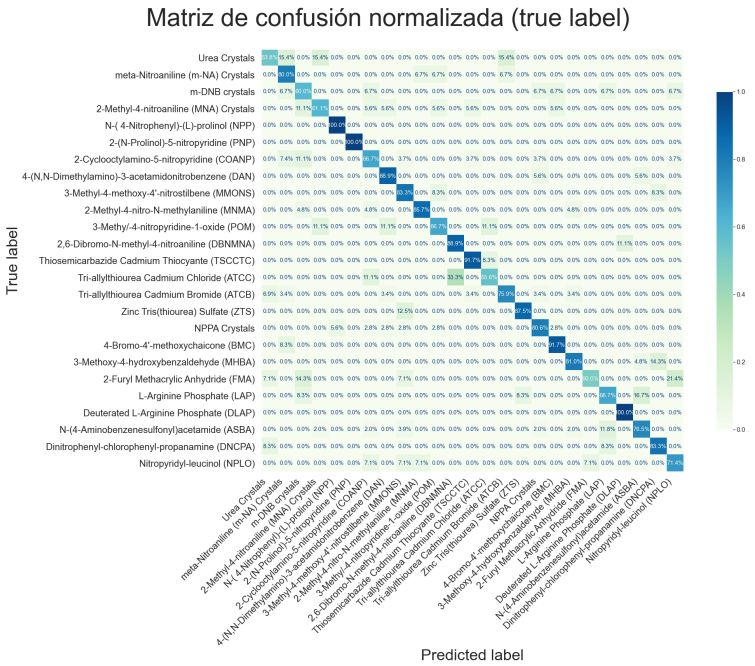


Figura B.4: Modelo entrenado con datos aumentados y evaluado con datos observados, normalizado. Elaboración Propia.

Figura B.5: Matriz de confusión para datos balanceados. Elaboración Propia.

Figura B.6: Matriz de confusión para datos balanceados, normalizado. Elaboración Propia.

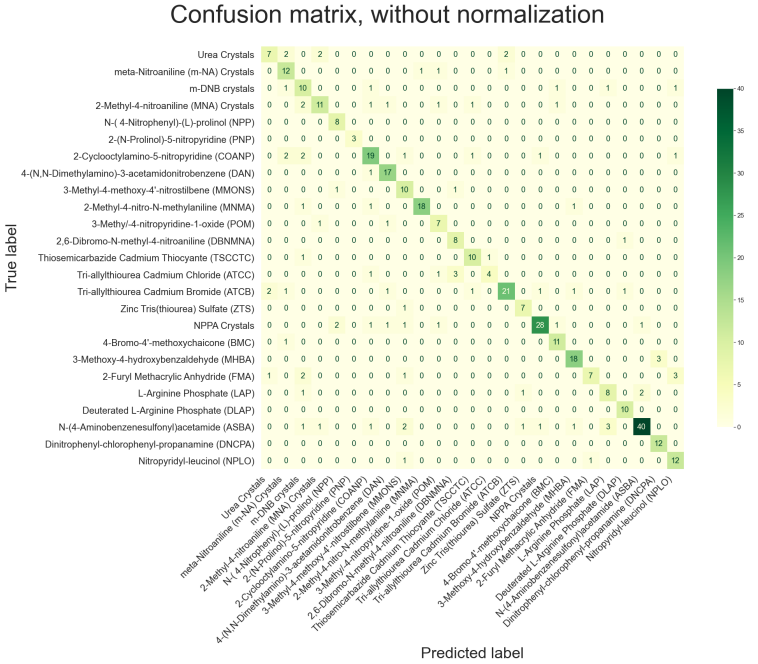


Figura B.7: Entrenado con datos balanceados, evaluado con datos observados. Elaboración Propia.

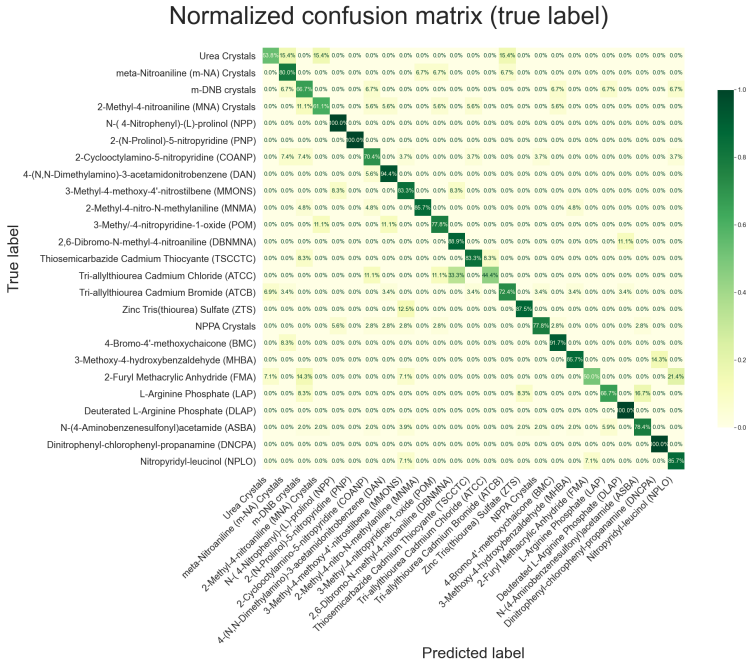


Figura B.8: Entrenado con datos balanceados, evaluado con datos observados, normalizado. Elaboración Propia.